D. LASSERAN

35 PROGRAMMES POUR ORIC 1+ATMOS

POCHE *informatique* 17

Collection

POCHE-INFORMATIQUE

35 PROGRAMMES POUR ORIC 1 ET ATMOS

par

Dominique LASSERAN

Diffusion :

EDITIONS TECHNIQUES ET SCIENTIFIQUES FRANÇAISES

2 à 12, rue de Bellevue, 75940 Paris Cedex 19

COLLECTION POCHE-INFORMATIQUE

1 - G. ISABEL, *50 programmes pour ZX 81*
2 - P. GUEULE, *Montages périphériques pour ZX 81*
3 - C. GALAIS, *Passeport pour Applesoft*
4 - R. BUSCH, *Passeport pour Basic*
5 - M. ROUSSELET, *Mathématiques sur ZX 81*
6 - C. GALAIS, *Passeport pour ZX 81*
7 - G. PROBST, *50 programmes pour Casio FX-702 et 801 P*
8 - G. PROBST, *60 programmes pour Casio PB 100*
9 - M. SAAL, *Utilitaires pour ZX 81*
10 - C. GALAIS, *Passeport pour Commodore 64*
11 - D. RANC, *Assembleur du TRS 80*
12 - D. LASSERAN, *30 programmes pour Commodore 64*
13 - G. ISABEL, *Du ZX 81 au Spectrum, 25 programmes*
14 - P. MELUSSON, *Initiation à la micro-informatique, le microprocesseur*
15 - G. PROBST, *40 programmes pour PB 700*
16 - C. GALAIS, *Passeport pour Basic TO 7 et TO 7-70*
17 - D. LASSERAN, *35 programmes pour Oric 1 et Atmos*

ISBN 2-85535-100-6 ISSN 0757-6730

SOMMAIRE

JEUX

VIE PRATIQUE

MATHÉMATIQUES

ASTRONOMIE

UTILITAIRES

PREFACE

Parmi les micro-ordinateurs familiaux récents, l'Oric est sans aucun doute celui qui présente le meilleur rapport performance/prix. Il possède : mémoire importante, haute résolution, générateur de sons et un Basic très complet.

Le but de cet ouvrage est de familiariser le lecteur, débutant ou averti, aux nombreuses possibilités de l'Oric, en lui donnant sous forme de programmes complets des exemples d'utilisation des commandes Basic.

Dans cet esprit, les 35 programmes sont présentés ainsi : tout d'abord le but du programme, ensuite son utilisation, puis un commentaire détaillé des différentes parties du listing, le listing lui-même, et enfin un exemple d'application ou une photographie.

On trouvera bien entendu des jeux de hasard et de réflexion, tels que *La tour infernale, Master Mind* ou *Bataille navale,* et des programmes plus « utiles », comme *Calendrier* ou *Calculatrice.* Les mathématiques sont présentes, avec entre autres un calcul de grandes factorielles et un algorithme d'intégration selon la méthode de Simpson. L'astronomie est également de la partie, avec notamment un programme pour déterminer les heures de lever et de coucher du soleil, et un pour convertir des coordonnées astronomiques, qui rendra service à l'astronome amateur. On découvrira enfin des programmes utilitaires tels que la redéfinition du jeu de caractères de l'Oric, ou un désassembleur pour ceux qui veulent découvrir le langage du microprocesseur 65Ø2.

Bien entendu, tous les programmes sont écrits dans le Basic standard de l'Oric, suivant des organigrammes structurés permettant une lecture aisée des listings, même pour le débutant.

Les programmes peuvent naturellement être utilisés tels quels, mais ils peuvent aussi servir, plus ou moins modifiés, de point de départ ou de sous-programmes à des programmes de plus grande envergure.

AVERTISSEMENT

La plupart des programmes présentés dans cet ouvrage tournent aussi bien sur Oric 1 que sur Oric Atmos. Cependant, certains programmes utilisent la fonction PRINT@, que l'Oric 1 ne possède pas. Ces programmes, repérés par le symbole ✱ placé sous le titre, peuvent toutefois tourner sur Oric 1 avec de légères modifications : il faut remplacer les instructions du type PRINT@ X, Y; A$ par PLOT X, Y, A$. Eventuellement certaines instructions du type PRINT A$ seront aussi à remplacer par PLOT X, Y, A$, où X et Y sont à déterminer, pour améliorer la présentation.

LOTO



Pour les enragés du Loto, ce programme donne un nouveau moyen de choisir les numéros à jouer. Bonne chance !

Utilisation.

Le programme affiche automatiquement les sept numéros à jouer.

Structure du programme.

Lignes 1Ø-7Ø : initialisations-affichage.
Lignes 8Ø-16Ø : tirage aléatoire de sept nombres différents compris entre 1 et 49.
Lignes 17Ø-21Ø : affichage des sept nombres.

Programme

```
10 REM  LOTO
20 PAPER 0:INK 2
30 CLS
40 PRINT:PRINT " Ce programme peut vous
rapporter"
50 PRINT "gros !"
60 PRINT " En effet, il vous donne les
7 numeros";
70 PRINT "que vous devez jouer au loto.
"
80 FOR I=1 TO 7
90 R=INT(RND(1)*49+1)
100 B=0
110 FOR J=0 TO I-1
120 IF R=A(J) THEN B=1
```

```
130 NEXT
140 IF B=1 GOTO 90
150 A(I)=R
160 NEXT
170 PRINT @ 2,11;CHR$(4);CHR$(27);"J"
180 FOR I=1 TO 7
190 PRINT @ 4+4*I,11;A(I)
200 NEXT
210 PRINT @ 2,20;CHR$(4)
```

Exemple

Ce programme peut vous rapporter gros !

En effet, il vous donne les 7 numeros que vous devez jouer au loto.

```
        20  19  13  7   32  47  25
```

CADAVRES EXQUIS



Ce jeu, très apprécié par les surréalistes, consiste à assembler des morceaux de phrases; on obtient ainsi des résultats grammaticalement corrects mais souvent surprenants et amusants par leur sens.

Utilisation

Il suffit de lancer le programme.

Structure du programme

Lignes 1Ø-7Ø : initialisations - affichage.
Lignes 8Ø-9Ø : attend une action de l'utilisateur pour continuer.
Ligne 1ØØ : N représente le nombre de morceaux de phrases connus de l'ordinateur.
Lignes 11Ø-15Ø : choisit par trois fois un morceau de phrase aléatoire. Affichage correspondant.
Lignes 16Ø-19Ø : demande pour une autre phrase.
Sous-programmes 2ØØ-21Ø, 3ØØ-31Ø, 4ØØ-41Ø : chacun de ces sous-programmes permet d'afficher le morceau de phrase de numéro J.

Programme

```
 10 REM  CADAVRES EXQUIS
 20 PAPER 0:INK 2
 30 CLS
 40 PRINT:PRINT "Amateurs de litterature
 experimentale,";
 50 PRINT "ce programme va vous plaire !
"
 60 PRINT:PRINT "Des phrases desopilante
s vont"
```

```
 70 PRINT "s'afficher sous vos yeux ebau
bis !"
 80 PRINT @ 2,23;"Appuyez sur une touche
 pour commencer."
 90 GET K$
 100 N=10
 110 PRINT:PRINT
 120 FOR I=1 TO 3
 130 J=INT(RND(1)*N+1)
 140 ON I GOSUB 200,300,400
 150 NEXT
 160 PRINT:PRINT "Une autre phrase ?"
 170 GET K$
 180 IF K$<>"N" GOTO 110
 190 END
 200 ON J GOTO 201,202,203,204,205,206,2
07,208,209,210
 201 PRINT "Une voiture ";:RETURN
 202 PRINT "Mon Oric ";:RETURN
 203 PRINT "L'ascenseur ";:RETURN
 204 PRINT "Une perceuse ";:RETURN
 205 PRINT "Le magnetophone ";:RETURN
 206 PRINT "Je ";:RETURN
 207 PRINT "Une vache ";:RETURN
 208 PRINT "Le pigeon ";:RETURN
 209 PRINT "Un jambon ";:RETURN
 210 PRINT "L'avion ";:RETURN
 300 ON J GOTO 301,302,303,304,305,306,3
07,308,309,310
 301 PRINT "mange ";:RETURN
 302 PRINT "roule sur ";:RETURN
 303 PRINT "pousse ";:RETURN
 304 PRINT "vole ";:RETURN
 305 PRINT "calcule ";:RETURN
 306 PRINT "casse ";:RETURN
 307 PRINT "derape sur ";:RETURN
 308 PRINT "tombe sur ";:RETURN
 309 PRINT "chante ";:RETURN
 310 PRINT "regarde ";:RETURN
```

```
400 ON J GOTO 401,402,403,404,405,406,4
07,408,409,410
401 PRINT "un agent de police.":RETURN
402 PRINT "des racines carrees.":RETURN
403 PRINT "la television.":RETURN
404 PRINT "le nuage.":RETURN
405 PRINT "un bonbon.":RETURN
406 PRINT "une enceinte.":RETURN
407 PRINT "le telephone.":RETURN
408 PRINT "le dictionnaire.":RETURN
409 PRINT "une poubelle.":RETURN
410 PRINT "le clocher.":RETURN
```

Exemple

```
Amateurs de litterature experimentale,
ce programme va vous plaire !
Des phrases desopilantes vont
s'afficher sous vos yeux ebaubis !
Une voiture roule sur le clocher.
Une autre phrase ?
Mon Oric regarde le dictionnaire.
Une autre phrase ?
L'avion tombe sur la television.
Une autre phrase ?
Le pigeon tombe sur un agent de police.
Une autre phrase ?
Un jambon tombe sur un agent de police.
Une autre phrase ?
L'ascenseur chante une poubelle.
Une autre phrase ?
```

CLAVIER-REFLEXE



Ce jeu d'observation et de réflexes va vous permettre de tester votre connaissance du clavier de l'Oric.

Utilisation

Une lettre de l'alphabet s'affiche aléatoirement sur l'écran. Vous devez appuyer le plus rapidement possible sur la touche correspondante du clavier de l'Oric. Cela peut vous sembler facile, mais attention ! L'écran n'est jamais effacé, si bien que quand une vingtaine de lettres y sont déjà éparpillées, il devient difficile de repérer où vient s'afficher la suivante. De plus, le jeu se déroule en temps limité : votre score correspond au nombre de lettres que vous avez reconnues durant ce temps.

Structure du programme

Lignes 1Ø-5Ø : initialisations - la ligne 3Ø sert à faire disparaître le curseur - SC représente le score, T est un compteur de temps.
Lignes 6Ø-8Ø : choisit des coordonnées aléatoires (X, Y) et une lettre aléatoire (A$).
Ligne 9Ø : affiche la lettre choisie sur l'écran.
Lignes 1ØØ-12Ø : cette boucle décompte le temps en attendant que le joueur appuie sur la bonne touche (KEY$=A$). La boucle s'arrête aussi si le temps est dépassé (T=Ø).
Ligne 13Ø : augmente le score si le joueur a bien répondu et reboucle pour une nouvelle tentative.
Lignes 14Ø-19Ø : le jeu est terminé - affichage du score - demande pour une nouvelle partie.

Programme

```
10 REM  CLAVIER-REFLEXE
20 PAPER 0:INK 2
```

```
 30 PRINT CHR$(17)
 40 CLS
 50 SC=0:T=5000
 60 X=INT(RND(1)*38+2)
 70 Y=INT(RND(1)*26)
 80 A$=CHR$(INT(RND(1)*26+65))
 90 PRINT @ X,Y;A$
 100 REPEAT
 110 T=T-1
 120 UNTIL KEY$=A$ OR T=0
 130 IF T>0 THEN SC=SC+1:GOTO 60
 140 CLS
 150 PRINT @ 8,10;"Votre score :";SC
 160 PRINT @ 8,14;"Voulez-vous rejouer ?
"
 170 GET K$
 180 IF K$<>"N" THEN PRINT @ 8,18;"Atten
tion !":WAIT 200:GOTO 40
 190 PRINT CHR$(17)
```

BATAILLE



Ce programme distribue les cartes au hasard, puis joue à la bataille contre vous.

Utilisation

Il suffit de lancer le programme.

Structure du programme

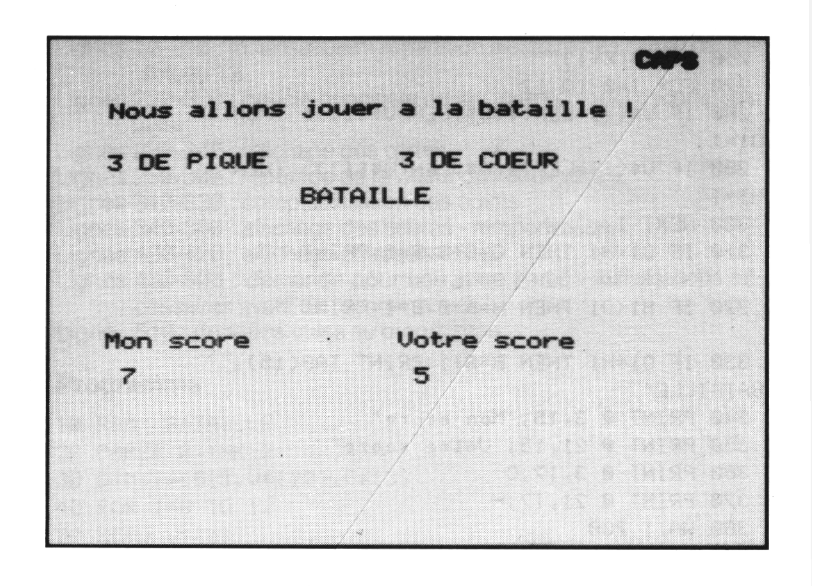
Lignes 10-210 : initialisations - répartition aléatoire des cartes dans le tableau T$.
Lignes 220-390 : boucle principale du jeu, que l'on peut décomposer ainsi :
Lignes 230-240 : affichage des cartes.
Lignes 250-300 : repérage de la valeur des deux cartes.
Lignes 310-330 : comptabilisation des points.
Lignes 340-380 : affichage des scores - temporisation.
Lignes 400-420 : affichage du résultat final.
Lignes 430-500 : demande pour une autre partie - initialisations nécessaires avant de recommencer.
Ligne 510 : données utiles au programme.

Programme

```
10 REM  BATAILLE
20 PAPER 0:INK 2
30 DIM T$(51),V$(12),C$(3)
40 FOR I=0 TO 12
50 READ V$(I)
60 NEXT
70 FOR J=0 TO 3
80 READ C$(J)
```

```
 90 NEXT
 100 CLS:PRINT CHR$(17)
 110 PRINT:PRINT " Nous allons jouer a l
a bataille !"
 120 FOR I=0 TO 12
 130 FOR J=0 TO 3
 140 K=INT(RND(1)*52)
 150 IF T$(K)="" THEN T$(K)=V$(I)+" DE "
+C$(J):GOTO 190
 160 K=K+1
 170 IF K=52 THEN K=0
 180 GOTO 150
 190 NEXT J
 200 NEXT I
 210 O=0:H=0:B=1
 220 FOR K=0 TO 50 STEP 2
 230 PRINT @ 3,5;T$(K);"              "
 240 PRINT @ 21,5;T$(K+1);"              "
 250 O$=T$(K)
 260 H$=T$(K+1)
 270 FOR I=0 TO 12
 280 IF V$(I)=LEFT$(O$,LEN(V$(I))) THEN
O1=I
 290 IF V$(I)=LEFT$(H$,LEN(V$(I))) THEN
H1=I
 300 NEXT I
 310 IF O1<H1 THEN O=O+B:B=1:PRINT "
                                       "
 320 IF H1<O1 THEN H=H+B:B=1:PRINT "
                                       "
 330 IF O1=H1 THEN B=B+1:PRINT TAB(15);"
BATAILLE"
 340 PRINT @ 3,15;"Mon score"
 350 PRINT @ 21,15;"Votre score"
 360 PRINT @ 3,17;O
 370 PRINT @ 21,17;H
 380 WAIT 200
 390 NEXT K
 400 IF O>H THEN PRINT @ 3,20;"J'ai gagn
e !"
```

```
410 IF O<H THEN PRINT @ 3,20;"Vous avez
gagne."
420 IF O=H THEN PRINT @ 3,20;"Match nul
!"
430 PRINT @ 3,25;"Voulez-vous rejouer ?
"
440 GET K$
450 IF K$="N" THEN PRINT CHR$(17):END
460 FOR K=0 TO 51
470 T$(K)=""
480 NEXT
490 CLS
500 GOTO 110
510 DATA AS,ROI,DAME,VALET,10,9,8,7,6,5
,4,3,2,PIQUE,COEUR,CARREAU,TREFLE
```



LE NOMBRE MYSTERIEUX

Vous devez découvrir un nombre choisi par votre Oric.

Utilisation

Vous devez choisir l'un des trois niveaux suivants :
Niveau 1 : le nombre mystérieux est compris entre 1 et 1ØØ.
Niveau 2 : le nombre mystérieux est compris entre 1 et 1 ØØØ.
Niveau 3 : le nombre mystérieux est compris entre 1 et 1Ø ØØØ.

Puis vous devez proposer un nombre. L'ordinateur vous indique alors s'il est trop grand ou trop petit, et ainsi de suite jusqu'à ce que vous ayez trouvé.

Structure du programme

Lignes 1Ø-3Ø : initialisations.
Lignes 4Ø-6Ø : introduction du niveau.
Ligne 7Ø : création aléatoire du nombre mystérieux NM.
Ligne 8Ø : introduction du nombre N.
Ligne 9Ø : teste si le joueur a gagné.
Lignes 1ØØ-15Ø : choisit aléatoirement l'un des cinq commentaires et l'affiche.
Lignes 16Ø-17Ø : affichage de la réponse.
Ligne 18Ø : incrémente la variable NC qui indique combien de nombres le joueur a déjà proposé.
Ligne 19Ø : retourne en ligne 8Ø (on est ici dans le cas où le joueur n'a pas trouvé le nombre mystérieux).
Lignes 2ØØ-24Ø : le joueur a gagné - affichage - demande pour une nouvelle partie.
Lignes 3ØØ-34Ø : données correspondant aux cinq commentaires pouvant être affichés.

Programme

```
 10 REM  LE NOMBRE MYSTERIEUX
 20 PAPER 0:INK 2
 30 CLS:NC=1
 40 PRINT "Niveau de difficulte (1,2,3)
?"
 50 GET K$
 60 IF K$<"1" OR K$>"3" GOTO 50
 70 NM=INT(RND(1)*10^(VAL(K$)+1)+1)
 80 PRINT:PRINT:INPUT "Votre nombre ";N
 90 IF N=NM GOTO 200
 100 C=INT(RND(1)*5)
 110 RESTORE
 120 FOR I=0 TO C
 130 READ C$
 140 NEXT
 150 PRINT:PRINT C$
 160 PRINT:PRINT "Votre nombre est trop
";
 170 IF N>NM THEN PRINT "grand." ELSE PR
INT "petit."
 180 NC=NC+1
 190 GOTO 80
 200 PRINT:PRINT "Ah ! Enfin ! Ce n'est
pas trop tot !"
 210 PRINT:PRINT "Vous avez trouve en";N
C;"coups."
 220 PRINT:PRINT:PRINT "Voulez-vous rejo
uer ?"
 230 GET K$
 240 IF K$<>"N" GOTO 30
 300 DATA "Bof ! Pas terrible."
 310 DATA "C'est lamentable."
 320 DATA "J'espere que vous allez faire
 mieux."
 330 DATA "Si vous continuez comme ca...
"
 340 DATA "Je n'ose pas repondre. Enfin.
.."
```

Exemple

```
Niveau de difficulte (1,2,3) ?

Votre nombre ? 500
Si vous continuez comme ca...
Votre nombre est trop grand.

Votre nombre ? 200
Si vous continuez comme ca...
Votre nombre est trop grand.

Votre nombre ? 120
J'espere que vous allez faire mieux.
Votre nombre est trop grand.

Votre nombre ? 40
Bof ! Pas terrible.
Votre nombre est trop petit.

Votre nombre ? 80
C'est lamentable.
Votre nombre est trop grand.

Votre nombre ? 68
Bof ! Pas terrible.
Votre nombre est trop grand.

Votre nombre ? 54
C'est lamentable.
Votre nombre est trop petit.

Votre nombre ? 60

Ah ! Enfin ! Ce n'est pas trop tot !

Vous avez trouve en 8 coups.

Voulez-vous rejouer ?
```

LA TOUR INFERNALE

Vous êtes dans une tour comportant un million de pièces : cent de long, cent de large et cent de haut. Vous venez d'apprendre qu'une bombe est cachée dans une des pièces, et vous êtes chargé de la trouver. Vous disposez d'un détecteur de bombe, affichant un nombre d'autant plus élevé (jusqu'à 1Ø ØØØ) que vous vous rapprochez d'elle. Cependant, les indications du détecteur varient de façon bizarre.

Utilisation

A chaque essai, vous devez introduire les trois coordonnées (longueur, largeur, hauteur, comprises entre Ø et 99).

Structure du programme

Lignes 1Ø-14Ø : initialisations-affichage.
Lignes 15Ø-18Ø : calcul de la position de la bombe.
Lignes 19Ø-23Ø : initialisations-affichage.
Lignes 24Ø-32Ø : boucle permettant l'affichage des résultats et l'introduction des coordonnées jusqu'à ce que le joueur perde ou gagne.
Lignes 33Ø-45Ø : affichage de commentaires.
Lignes 47Ø-48Ø : demande pour une autre partie.

Programme

```
 10 REM  LA TOUR INFERNALE
 20 PAPER 0:INK 2
 30 CLS:PRINT:PRINT:PRINT:PRINT
 40 PRINT " Vous etes dans une tour qui
comporte"
 50 PRINT "un million de pieces.La tour
fait 100"
```

```
 60 PRINT "pieces de haut, 100 de long e
t 100 de"
 70 PRINT "large, numerotees de 0 a 99."
:PRINT
 80 PRINT " Une bombe est cachee dans un
e piece"
 90 PRINT "et vous devez la desamorcer."
:PRINT
 100 PRINT " Vous disposez d'un detecteu
r qui"
 110 PRINT "affiche un nombre d'autant p
lus grand"
 120 PRINT "que vous vous rapprochez de
la bombe.":PRINT
 130 PRINT " Vous avez 20 coups pour reu
ssir."
 140 WAIT 2000
 150 XB=INT(RND(1)*100)
 160 YB=INT(RND(1)*100)
 170 ZB=INT(RND(1)*100)
 180 IF XB=0 AND YB=0 AND ZB=0 GOTO 160
 190 XJ=0:YJ=0:ZJ=0:NC=0
 200 CLS:PRINT
 210 PRINT " Signal";TAB(11);"Lon";TAB(1
5);"Lar";TAB(19);"Hau";
 220 PRINT TAB(23);"Essai";TAB(29);"Coor
d."
 230 PRINT
 240 REPEAT
 250 PRINT 10000-ABS((XB/100+YB+ZB*100)-
(XJ/100+YJ+ZJ*100));
 260 PRINT TAB(11);XJ;TAB(15);YJ;TAB(19)
;ZJ;TAB(23);NC;TAB(29);
 270 INPUT XJ,YJ,ZJ
 280 NC=NC+1
 290 F= XJ<0 OR XJ>99 OR XJ<>INT(XJ) OR
YJ<0 OR YJ>99 OR YJ<>INT(YJ)
 300 F=F OR ZJ<0 OR ZJ>99 OR ZJ<>INT(ZJ)
 OR NC>20
 310 G= XJ=XB AND YJ=YB AND ZJ=ZB
```

```
320 UNTIL F OR G
330 IF G THEN PRINT:PRINT "Vous avez de
sactive la bombe en"
340 IF G THEN PRINT NC;"tentatives.":GO
TO 460
350 IF ZJ<0 OR ZJ>99 THEN PRINT:PRINT "
Vous etes maintenant a"
360 IF ZJ<0 THEN PRINT -3*ZJ;"metres so
us terre.":GOTO 420
370 IF ZJ>99 THEN PRINT 3*ZJ;"metres du
 sol.":GOTO 420
380 F= XJ<0 OR XJ>99 OR YJ<0 OR YJ>99
390 IF F THEN PRINT:PRINT "Vous etes to
mbe de l'etage No";ZJ:GOTO 420
400 F= XJ<>INT(XJ) OR YJ<>INT(YJ) OR ZJ
<>INT(ZJ)
410 IF F THEN PRINT:PRINT "Vous etes co
ince dans un mur."
420 EXPLODE
430 PRINT:PRINT "Tout a explose !"
440 PRINT:PRINT "La bombe etait situee
aux coordonnees:"
450 PRINT "Long,Larg,Haut =";XB;YB;ZB
460 PRINT:PRINT "Voulez-vous rejouer ?"
470 GET K$
480 IF K$<>"N" GOTO 150
```

Exemple

Vous etes dans une tour qui comporte un million de pieces.La tour fait 100 pieces de haut, 100 de long et 100 de large, numerotees de 0 a 99.

Une bombe est cachee dans une piece et vous devez la desamorcer.

Vous disposez d'un detecteur qui affiche un nombre d'autant plus grand que vous vous rapprochez de la bombe.

```
Vous avez 20 coups pour reussir.

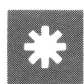
  Signal   Lon  Lar  Hau  Essai  Coord.

  7423.11  0    0    0    0      ? 50,50,25
  9973.61  50   50   25   1      ? 45,60,28
  9716.44  45   60   28   2      ? 25,45,23
  9768.36  25   45   23   3      ? 40,70,25
  9993.51  40   70   25   4      ? 60,72,25
  9995.71  60   72   25   5      ? 70,75,25
  9998.81  70   75   25   6      ? 80,77,25
  9999.09  80   77   25   7      ? 10,76,25
  9999.21  10   76   25   8      ? 30,76,25
  9999.41  30   76   25   9      ? 50,76,25
  9999.61  50   76   25   10     ? 90,76,25
  9999.99  90   76   25   11     ? 91,76,25
  9999.98  91   76   25   12     ? 89,76,25

Vous avez desactive la bombe en
 13 tentatives.

 Voulez-vous rejouer ?
```

MASTER MIND



Vous devez découvrir un code secret créé par l'ordinateur.

Utilisation

Il y a deux niveaux de jeu ; vous en choisissez un au début de la partie.

A chaque coup, vous devez introduire un code formé de chiffres, sans laisser d'espace ni mettre de virgule entre ces chiffres.

Structure du programme

Lignes 1Ø-5Ø : initialisations.
Lignes 6Ø-8Ø : création aléatoire du code secret.
Lignes 9Ø-34Ø : boucle principale du jeu, qui se décompose ainsi :
Lignes 11Ø-18Ø : introduction, vérification, affichage du coup joué.
Lignes 19Ø-24Ø : calcul du nombre de chiffres mal placés.
Lignes 25Ø-3ØØ : calcul du nombre de chiffres bien placés.
Lignes 31Ø-32Ø : affichage de la réponse.
Ligne 33Ø : permet d'arrêter la boucle si le joueur a trouvé le code.
Lignes 35Ø-39Ø : affichage du code secret.
Lignes 4ØØ-44Ø : affichage - demande pour une autre partie.
Sous-programme 5ØØ-67Ø : affichage des instructions - initialisations.
Sous-programme 7ØØ-86Ø : choix du niveau - initialisations.

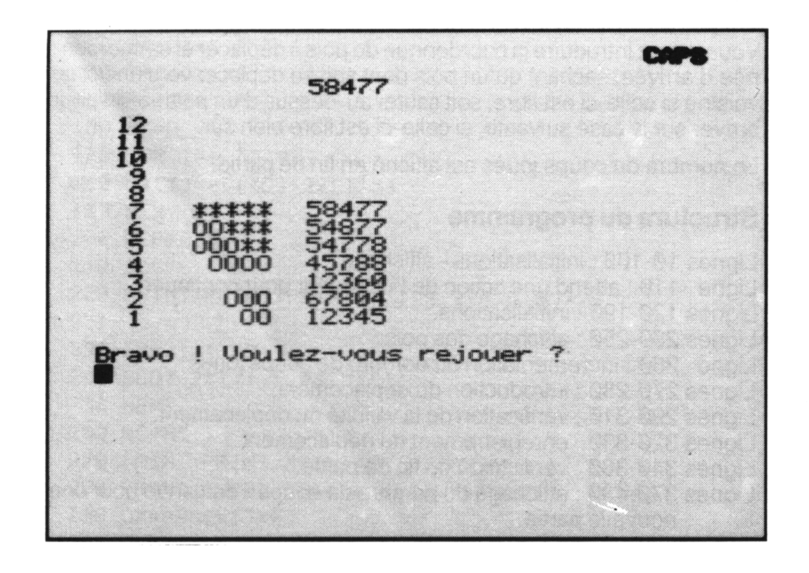
Programme

```
10 REM  MASTERMIND
20 PAPER 0:INK 2
30 CLS
40 GOSUB 500
50 GOSUB 700
60 FOR I=1 TO F
70 C(I)=INT(RND(1)*(C+1))
```

```
80 NEXT
90 FOR P=1 TO T
100 PRINT @ 2,16;"Votre coup ";
110 INPUT K$
120 IF LEN(K$)<>F GOTO 100
130 E=0
140 FOR I=1 TO F
150 IF MID$(K$,I,1)<"0" OR MID$(K$,I,1)
>RIGHT$(STR$(C),1) THEN E=1:I=F
160 NEXT
170 IF E=1 GOTO 100
180 PRINT @ 15,15-P;K$
190 B=0
200 FOR I=1 TO F
210 D(I)=C(I)
220 G(I)=VAL(MID$(K$,I,1))
230 IF G(I)=D(I) THEN B=B+1:G(I)=10:D(I
)=9
240 NEXT
250 W=0
260 FOR I=1 TO F
270 FOR J=1 TO F
280 IF G(J)=D(I) THEN W=W+1:G(J)=10:D(I
)=9
290 NEXT J
300 NEXT I
310 R$=LEFT$("OOOOO",W)+LEFT$("*****",B
)
320 PRINT @ 13-LEN(R$),15-P;R$
330 IF B=F THEN P=T
340 NEXT P
350 K$=""
360 FOR I=1 TO F
370 K$=K$+RIGHT$(STR$(C(I)),1)
380 NEXT
390 PRINT @ 15,1;K$
400 IF B=F THEN PRINT @ 2,16;"Bravo !";
ELSE PRINT @ 2,16;"Perdu !";
410 PRINT " Voulez-vous rejouer ?"
420 GET K$
```

```
430 IF K$<>"N" GOTO 50
440 END
500 DIM G(5),C(5),D(5)
510 CLS
520 PRINT @ 10,1;"MASTER MIND"
530 PRINT @ 3,5;"Vous devez trouver le
code secret que";
540 PRINT " j'ai choisi en utilisant 4
chiffres de"
550 PRINT "0 a 6 pour le MINI MASTER MI
ND, ou"
560 PRINT "bien 5 chiffres de 0 a 8 pou
r le "
570 PRINT "SUPER MASTER MIND. Vous deve
z trouver"
580 PRINT "la solution en 6 ou 12 coups
 selon le"
590 PRINT "cas. Je repondrai a vos essa
is de la"
600 PRINT "maniere suivante :"
610 PRINT:PRINT " * : un de vos chiffre
s se trouve dans"
620 PRINT "      mon code et il est bien
 place."
630 PRINT:PRINT " O : un de vos chiffre
s se trouve dans"
640 PRINT "      mon code mais il est ma
l place."
650 PRINT:PRINT:PRINT "Appuyez sur une
touche pour continuer"
660 GET K$
670 RETURN
700 CLS
710 PRINT @ 10,1;"MASTER MIND"
720 PRINT @ 3,5;"Quelle variante prefer
ez-vous ?"
730 PRINT:PRINT "  1  MINI MASTER MIND"
740 PRINT:PRINT "  2  SUPER MASTER MIND
"
750 GET K$
```

```
 760 IF K$="1" THEN F=4:C=6:T=6:GOTO 790
 770 IF K$="2" THEN F=5:C=8:T=12:GOTO 79
0
 780 GOTO 750
 790 CLS
 800 PRINT @ 15,1;LEFT$("@@@@@",F)
 810 PRINT
 820 FOR I=12 TO 1 STEP -1
 830 IF I>T THEN PRINT:GOTO 850
 840 PRINT RIGHT$("  "+STR$(I),3)
 850 NEXT
 860 RETURN
```



LES POIS SAUTEURS

Quatre pois « * » et quatre pois « O » se font face. Le jeu consiste à inverser leurs positions de part et d'autre du centre.

Utilisation

Vous devez introduire la coordonnée du pois à déplacer et sa coordonnée d'arrivée, sachant qu'un pois peut soit se déplacer vers une case voisine si celle-ci est libre, soit sauter au-dessus d'un pois voisin pour arriver sur la case suivante, si celle-ci est libre bien sûr.

Le nombre de coups joués est affiché en fin de partie.

Structure du programme

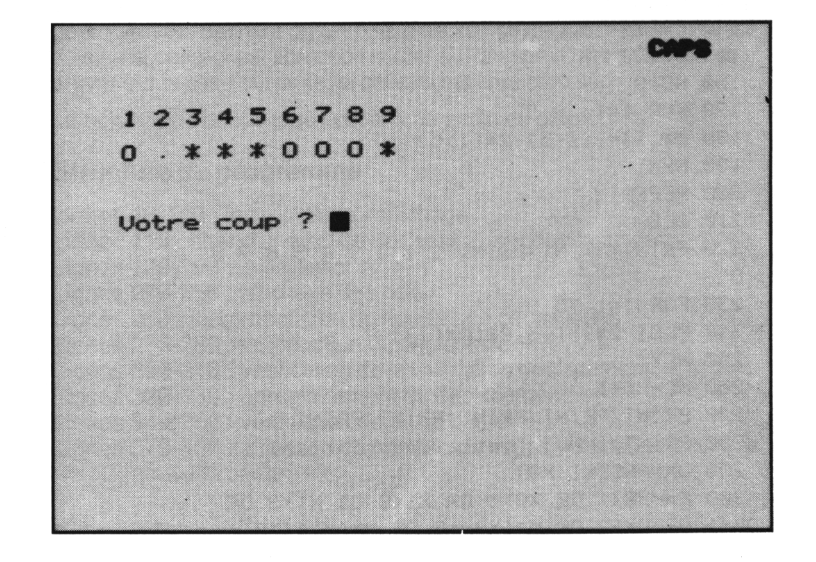
Lignes 10-100 : initialisations - affichage.
Ligne 110 : attend une action de l'utilisateur pour continuer.
Lignes 120-190 : initialisations.
Lignes 200-250 : affichage des pois.
Ligne 260 : incrémentation du nombre de coups joués.
Lignes 270-280 : introduction du déplacement.
Lignes 290-310 : vérification de la validité du déplacement.
Lignes 320-330 : enregistrement du déplacement.
Lignes 340-360 : vérification de fin de partie.
Lignes 370-430 : affichage du nombre de coups - demande pour une nouvelle partie.

Programme

```
  10 REM  LES POIS SAUTEURS
  20 PAPER 0:INK 2:CLS
  30 PRINT:PRINT " Huit pois sauteurs son
t alignes de"
```

```
 40 PRINT "part et d'autre du centre. Vo
us devez"
 50 PRINT "inverser leurs positions resp
ectives."
 60 PRINT:PRINT " Les pois peuvent se de
placer vers une"
 70 PRINT "case voisine si celle-ci est
libre, ou"
 80 PRINT "bien sauter par dessus un aut
re pois"
 90 PRINT "vers une case libre."
 100 PRINT:PRINT:PRINT:PRINT " Appuyez s
ur une touche pour jouer."
 110 GET K$
 120 DIM P$(2),M%(9)
 130 P$(0)="."
 140 P$(1)="*"
 150 P$(2)="O"
 160 NC=0
 170 FOR I=1 TO 9
 180 M%(I)=-(I<5)-2*(I>5)
 190 NEXT
 200 REPEAT
 210 CLS
 220 PRINT:PRINT:PRINT " 1 2 3 4 5 6 7 8
 9"
 230 FOR I=1 TO 9
 240 PLOT 2*I+1,5,P$(M%(I))
 250 NEXT
 260 NC=NC+1
 270 PRINT:PRINT:PRINT:PRINT:PRINT
 280 PRINT:INPUT "Votre coup ";X0,X1
 290 DX=ABS(X1-X0)
 300 F= X0<1 OR X0>9 OR X1<0 OR X1>9 OR
DX<1 OR DX>2 OR M%(X0)=0 OR M%(X1)<>0
 310 IF F THEN PRINT:PRINT "Deplacement
interdit ! Recommencez.":GOTO 280
 320 M%(X1)=M%(X0)
 330 M%(X0)=0
 340 T1=M%(1)+M%(2)+M%(3)+M%(4)
```

```
 350 T2=M%(6)+M%(7)+M%(8)+M%(9)
 360 UNTIL T1=8 AND T2=4
 370 CLS
 380 PRINT:PRINT:PRINT " 1 2 3 4 5 6 7 8
 9"
 390 PRINT:PRINT:PRINT " O O O O . * * *
 *"
 400 PRINT:PRINT:PRINT:PRINT "Vous avez
gagne en";NC;"coups."
 410 PRINT:PRINT "Voulez-vous rejouer ?"
 420 GET K$
 430 IF K$<>"N" GOTO 160
```



RAYON X

Des atomes d'impuretés se sont introduits dans un cristal lors de sa formation. Heureusement, un système optique utilisant les rayons X permet de les localiser. La méthode consiste à envoyer un rayon X dans le cristal, suivant une direction donnée, et à analyser le rayon émergent. Quatre cas peuvent se présenter :

1 - le rayon est absorbé lorsqu'il rencontre un atome ;

2 - le rayon est dévié de 90° lorsqu'il arrive sur une case diagonale du voisinage immédiat d'un atome ;

3 - le rayon est réfléchi lorsqu'il arrive entre deux atomes séparés par une seule case, ou lorsque l'atome est juste au bord du cristal et que le rayon est envoyé sur une case voisine de l'atome ;

4 - dans les autres cas, le rayon n'est pas dévié.

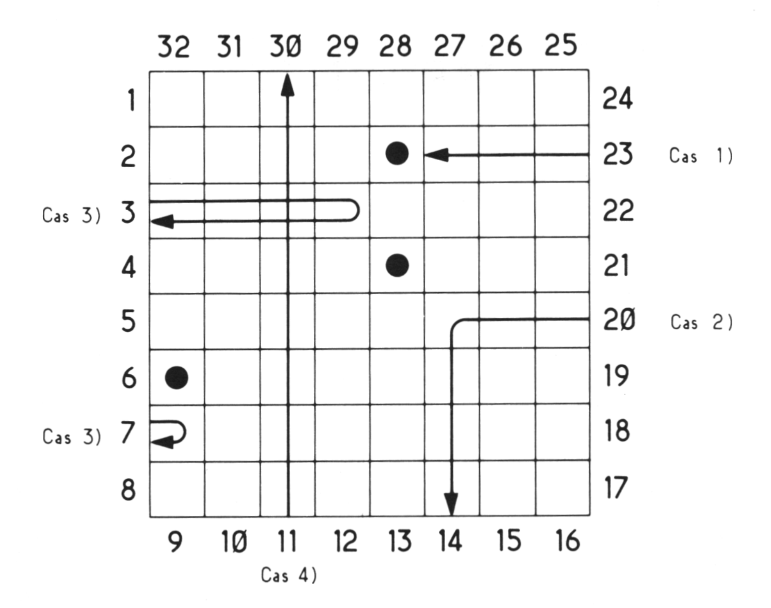
Utilisation

Vous choisissez tout d'abord le niveau de difficulté en introduisant le nombre d'atomes souhaité. Ensuite, vous introduisez le numéro de ligne ou de colonne (entre 1 et 32) où vous voulez envoyer le rayon X, puis l'ordinateur répond. Lorsque vous avez découvert les atomes, appuyez sur « Ø » ; introduisez alors les coordonnées des atomes, sous la forme « ligne, colonne », les numéros de ligne et de colonne étant compris entre 1 et 8.

Le score est comptabilisé de la manière suivante :

– 1 point pour un rayon réfléchi ou absorbé ;
– 2 points pour les rayons émergents ;
– 5 points pour chaque atome non trouvé.

Le but du jeu est évidemment de réaliser le plus petit score possible.



Structure du programme

Lignes 10-160 : initialisations - affichage - introduction.
Lignes 170-210 : calcul aléatoire des positions des atomes.
Ligne 220 : initialisations.
Lignes 230-510 : boucle principale - calcul de la trajectoire du rayon dans les différents cas.
Lignes 520-580 : introduction des résultats.
Lignes 590-660 : affichage du cristal.
Lignes 670-690 : affichage du score.
Lignes 700-720 : demande pour une autre partie.

Programme

```
  10 REM  RAYON X
  20 PAPER 0:INK 2
  30 CLS:PRINT:PRINT " Des atomes d'impur
etes se sont"
```

```
 40 PRINT "introduits dans un cristal, e
t vous"
 50 PRINT "etes charge de les localiser.
"
 60 PRINT:PRINT " Pour cela, vous envoye
z a l'interieur";
 70 PRINT "un rayon X.Celui-ci peut etre
 devie ou"
 80 PRINT "absorbe, et vous devez deduir
e de ces"
 90 PRINT "resultats la position des ato
mes."
 100 PRINT:PRINT " Combien d'atomes voul
ez-vous"
 110 INPUT "rechercher ";NB:PRINT
 120 FOR I=0 TO 9
 130 FOR J=0 TO 9
 140 C(I,J)=0
 150 NEXT J
 160 NEXT I
 170 FOR I=1 TO NB
 180 X=1+INT(8*RND(1))
 190 Y=1+INT(8*RND(1))
 200 IF C(X,Y)=0 THEN C(X,Y)=1 ELSE GOTO
 180
 210 NEXT
 220 A=0:SC=0
 230 INPUT "Rayon ";R
 240 IF R<1 GOTO 520
 250 IF R<9 THEN X=0:Y=R:IX=1:IY=0:GOTO
320
 260 IF R<17 THEN X=R-8:Y=9:IX=0:IY=-1:G
OTO 320
 270 IF R<25 THEN X=9:Y=25-R:IX=-1:IY=0:
GOTO 320
 280 IF R<33 THEN X=33-R:Y=0:IX=0:IY=1:G
OTO 320
 290 PRINT "Erreur ! Vous devez donner u
n nombre"
```

```
 300 PRINT "compris entre 0 et 32.Recomm
encez."
 310 GOTO 230
 320 XS=X+IX
 330 YS=Y+IY
 340 IF IX=0 THEN XG=XS-1:XD=XS+1:YG=YS:
YD=YS:GOTO 370
 350 XG=XS:YG=YS-1
 360 XD=XS:YD=YS+1
 370 IF C(XS,YS)=1 THEN PRINT "Le rayon
est absorbe.":SC=SC+1:GOTO 230
 380 IF C(XD,YD)=0 AND C(XG,YG)=0 THEN X
=XS:Y=YS:GOTO 430
 390 IF C(XD,YD)=1 THEN DI=-1
 400 IF C(XG,YG)=1 THEN DI=1
 410 IF IX=0 THEN IX=DI:IY=0:GOTO 430
 420 IF IY=0 THEN IX=0:IY=DI
 430 IF X<1 THEN RE=Y:GOTO 480
 440 IF X>8 THEN RE=25-Y:GOTO 480
 450 IF Y<1 THEN RE=33-X:GOTO 480
 460 IF Y>8 THEN RE=X+8:GOTO 480
 470 GOTO 320
 480 IF RE<>R THEN PRINT "Le rayon";R;"s
ort en";RE:SC=SC+2:GOTO 230
 490 PRINT "Le rayon est reflechi."
 500 SC=SC+1
 510 GOTO 230
 520 PRINT:PRINT " Donnez la position de
s atomes sous"
 530 PRINT "la forme (colonne, ligne):"
 540 FOR I=1 TO NB
 550 PRINT:PRINT " Atome numero";I;
 560 INPUT C,L
 570 IF C(C,L)=1 THEN A=A+1 ELSE SC=SC+5
 580 NEXT
 590 PRINT
 600 FOR J=1 TO 8
 610 PRINT "   ";
 620 FOR I=1 TO 8
```

```
 630 IF C(I,J)=0 THEN PRINT " ."; ELSE P
RINT " *";
 640 NEXT I
 650 PRINT:PRINT
 660 NEXT J
 670 PRINT:PRINT:PRINT " Vous avez decou
vert";A;"des";NB
 680 PRINT "atomes."
 690 PRINT:PRINT " Votre score est ";SC
 700 PRINT:PRINT " Voulez-vous rejouer ?
"
 710 GET K$
 720 IF K$<>"N" GOTO 100
```

Exemple

```
 Des atomes d'impuretes se sont
introduits dans un cristal, et vous
etes charge de les localiser.

 Pour cela, vous envoyez a l'interieur
un rayon X.Celui-ci peut etre devie ou
absorbe, et vous devez deduire de ces
resultats la position des atomes.

 Combien d'atomes voulez-vous
rechercher ?  3

Rayon ?  2
Le rayon 2 sort en 31
Rayon ?  5
Le rayon 5 sort en 27
Rayon ?  7
Le rayon est reflechi.
Rayon ?  10
Le rayon 10 sort en 4
Rayon ?  0

 Donnez la position des atomes sous
la forme (colonne, ligne):
```

```
 Atome numero 1 ? 3,3

 Atome numero 2 ? 7,6

 Atome numero 3 ? 7,8

    . . . . . . . .
    . . . . . . . .
    . . * . . . . .
    . . . . . . . .
    . . . . . . . .
    . . . . . . * .
    . . . . . . . .
    . . . . . . * .

 Vous avez decouvert 3 des 3
atomes.

 Votre score est  7

 Voulez-vous rejouer ?
```

BATAILLE NAVALE



Vous allez jouer à la bataille navale contre votre Oric.

Utilisation

L'ordinateur dispose secrètement un vaisseau de quatre cases de long, deux de trois cases, trois de deux cases et quatre de une case.

A chaque tour, vous devez indiquer la case où vous désirez tirer en précisant d'abord la lettre, puis le chiffre (ex. : A9, B7).

L'ordinateur répond par un bruit et place sur une grille un « O » si vous n'avez fait aucun dégât, ou bien une « * » si vous avez touché un de ses vaisseaux.

Structure du programme

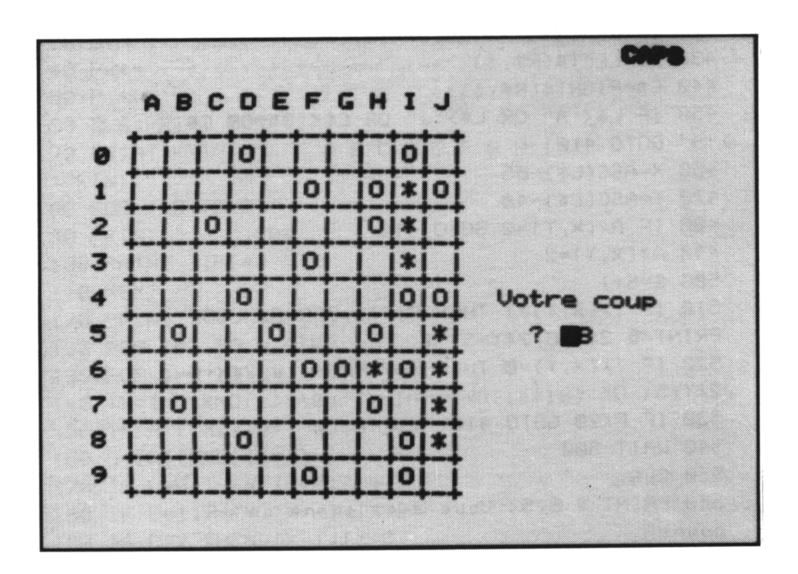
Lignes 1Ø-12Ø : initialisations - affichage d'une grille.
Lignes 13Ø-35Ø : place aléatoirement les vaisseaux dans le tableau T%.
Lignes 4ØØ-48Ø : introduction et vérification du coup.
Ligne 49Ø : indique pour plus tard que ce coup a été joué.
Ligne 5ØØ : incrémente S qui représente le nombre de coups joués.
Lignes 51Ø-52Ø : teste si ce coup touche un vaisseau ou non.
Ligne 53Ø : retourne à l'introduction du coup tant que tous les vaisseaux n'ont pas été détruits.
Lignes 54Ø-59Ø : affichage du nombre de coups joués - demande pour une nouvelle partie.
Lignes 6ØØ-66Ø : réinitialise les variables avant de rejouer.

Programme

```
10 REM  BATAILLE NAVALE
20 PAPER 0:INK 2
```

```
 30 DIM T%(9,9),A%(9,9)
 40 P0$="+-+-+-+-+-+-+-+-+-+-+"
 50 P1$="¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦ ¦"
 60 CLS:PRINT CHR$(17)
 70 PRINT:PRINT "     A B C D E F G H I J
":PRINT
 80 FOR I=0 TO 9
 90 PRINT "   ";P0$
 100 PRINT I;P1$
 110 NEXT
 120 PRINT "   ";P0$
 130 FOR I=3 TO 0 STEP -1
 135 FOR J=0 TO 3-I
 140 TX=INT(RND(1)*10):TY=INT(RND(1)*10)
 150 IF A%(TX,TY)=1 GOTO 140
 160 D=INT(RND(1)*4)
 170 IF D=0 THEN IX=0:IY=-1
 180 IF D=1 THEN IX=-1:IY=0
 190 IF D=2 THEN IX=1:IY=0
 200 IF D=3 THEN IX=0:IY=1
 210 QX=TX+I*IX:QY=TY+I*IY
 220 IF QX<0 OR QX>9 OR QY<0 OR QY>9 GOT
O 140
 230 IF A%(QX,QY)=1 GOTO 140
 240 FOR K=0 TO I
 250 T%(TX+K*IX,TY+K*IY)=1
 260 NEXT
 270 FOR K=-1 TO I+1
 280 FOR L=-1 TO 1
 290 X=TX+K*IX-L*IY
 300 Y=TY+K*IY+L*IX
 310 IF X>=0 AND X<=9 AND Y>=0 AND Y<=9
THEN A%(X,Y)=1
 320 NEXT L
 330 NEXT K
 340 NEXT J
 350 NEXT I
 400 PRINT @ 28,13;"Votre coup"
 410 PRINT @ 28,15;"  ";CHR$(17);:INPUT
R$:PRINT CHR$(17)
```

```
420 IF LEN(R$)<>2 GOTO 410
430 L$=LEFT$(R$,1)
440 C$=RIGHT$(R$,1)
450 IF L$<"A" OR L$>"J" OR C$<"0" OR C$
>"9" GOTO 410
460 X=ASC(L$)-65
470 Y=ASC(C$)-48
480 IF A%(X,Y)=2 GOTO 410
490 A%(X,Y)=2
500 S=S+1
510 IF T%(X,Y)=1 THEN P=P+1:ZAP:EXPLODE
PRINT @ 2*X+6,2*Y+5;"*"
520 IF T%(X,Y)=0 THEN ZAP:PRINT @ 2*X+6
,2*Y+5;"O"
530 IF P<20 GOTO 410
540 WAIT 500
550 CLS
560 PRINT @ 5,5;"Vous avez gagne en";S;
"coups"
570 PRINT @ 5,7;"Voulez-vous rejouer ?"
580 GET K$
590 IF K$="N" THEN PRINT CHR$(17):END
600 FOR I=0 TO 9
610 FOR J=0 TO 9
620 T%(I,J)=0:A%(I,J)=0
630 NEXT J
640 NEXT I
650 S=0:P=0
660 CLS:PRINT
670 GOTO 70
```


CAPS
A B C D E F G H I J
0
1
2
3
4
5
6
7
8
9
Votre coup
? B

SIMON



Dans ce jeu de mémoire, vous devez retrouver une séquence à la fois visuelle et musicale que votre Oric a créée.

Utilisation

A chaque tour, l'Oric vous fait entendre une séquence de notes et affiche en même temps la touche (1, 2, 3 ou 4) que vous devez enfoncer pour reproduire la même note. Vous devez ensuite répéter cette séquence grâce aux touches 1, 2, 3 ou 4, et appuyer sur la barre d'espacement quand vous avez fini. Si votre réponse est bonne, vous rejouez immédiatement avec la même séquence plus une note supplémentaire. Le jeu débute avec une séquence d'une note.

Structure du programme

Lignes 1Ø-4Ø : initialisations.
Lignes 5Ø-7Ø : affichage des scores.
Lignes 8Ø-9Ø : annonce le début du jeu.
Lignes 1ØØ-11Ø : la chaîne A$, qui contient la séquence, est augmentée d'une note aléatoire, déterminée par C.
Lignes 12Ø-15Ø : permet de faire écouter la séquence.
Lignes 16Ø-22Ø : affichage - introduction de la réponse (R$).
Ligne 23Ø : teste la réponse — si elle est bonne, on augmente le score et on recommence.
Lignes 24Ø-28Ø : affichage - demande pour une autre partie.
Sous-programme 1ØØØ-1Ø6Ø : affiche 1 en double hauteur, et émet un *do* grave.
Sous-programme 11ØØ-116Ø : affiche 2 en double hauteur, et émet un *mi.*
Sous-programme 12ØØ-126Ø : affiche 3 en double hauteur, et émet un *sol.*
Sous-programme 13ØØ-136Ø : affiche 4 en double hauteur, et émet un *do* aigu.

Programme

```
 10 REM  SIMON
 20 PAPER 0:INK 2:PRINT CHR$(17)
 30 MS=0
 40 SC=0:A$=""
 50 CLS
 60 PRINT @ 3,1;"Meilleur score :";MS
 70 PRINT @ 3,3;"Votre score    :";SC
 80 PRINT @ 3,20;"ATTENTION !"
 90 WAIT 100
 100 C=INT(RND(1)*4+1)
 110 A$=A$+RIGHT$(STR$(C),1)
 120 FOR I=1 TO LEN(A$)
 130 C=VAL(MID$(A$,I,1))
 140 ON C GOSUB 1000,1100,1200,1300
 150 NEXT
 160 PRINT @ 3,20;"Votre reponse ?"
 170 R$=""
 180 REPEAT
 190 GET K$
 200 C=VAL(K$)
 210 IF C>0 AND C<5 THEN R$=R$+K$:ON C G
OSUB 1000,1100,1200,1300
 220 UNTIL K$="  "
 230 IF R$=A$ THEN SC=SC+1:MS=MS-(SC>MS)
:GOTO 50
 240 PRINT @ 3,20;"C'est faux ! Voulez-v
ous rejouer ?"
 250 GET K$
 260 IF K$<>"N" GOTO 40
 270 PRINT CHR$(17)
 280 END
 1000 PRINT @ 3,11;CHR$(4);CHR$(27);"J"
 1010 PRINT @ 7,11;"1";CHR$(4)
 1020 MUSIC 1,3,1,15
 1030 WAIT 30
 1040 PRINT @ 7,11;CHR$(4);" ";CHR$(4)
 1050 MUSIC 1,0,1,0
 1060 RETURN
```

```
1100 PRINT @ 3,11;CHR$(4);CHR$(27);"J"
1110 PRINT @ 11,11;"2";CHR$(4)
1120 MUSIC 1,3,5,15
1130 WAIT 30
1140 PRINT @ 11,11;CHR$(4);" ";CHR$(4)
1150 MUSIC 1,0,1,0
1160 RETURN
1200 PRINT @ 3,11;CHR$(4);CHR$(27);"J"
1210 PRINT @ 15,11;"3";CHR$(4)
1220 MUSIC 1,3,8,15
1230 WAIT 30
1240 PRINT @ 15,11;CHR$(4);" ";CHR$(4)
1250 MUSIC 1,0,1,0
1260 RETURN
1300 PRINT @ 3,11;CHR$(4);CHR$(27);"J"
1310 PRINT @ 19,11;"4";CHR$(4)
1320 MUSIC 1,4,1,15
1330 WAIT 30
1340 PRINT @ 19,11;CHR$(4);" ";CHR$(4)
1350 MUSIC 1,0,1,0
1360 RETURN
```

CHENILLE



Vous déplacez une charmante chenille sur l'écran. Elle doit avaler autant de points affichés que possible. Mais attention ! Plus la chenille mange, plus elle dépose... d'excréments derrière elle ; dès qu'elle heurte quelque chose « qui n'est pas de la bonne nourriture », vous avez perdu.

Utilisation

Vous déplacez la chenille grâce aux quatre touches de déplacement du curseur.

Le score affiché en fin de partie correspond à la somme des points dévorés par la chenille.

Structure du programme

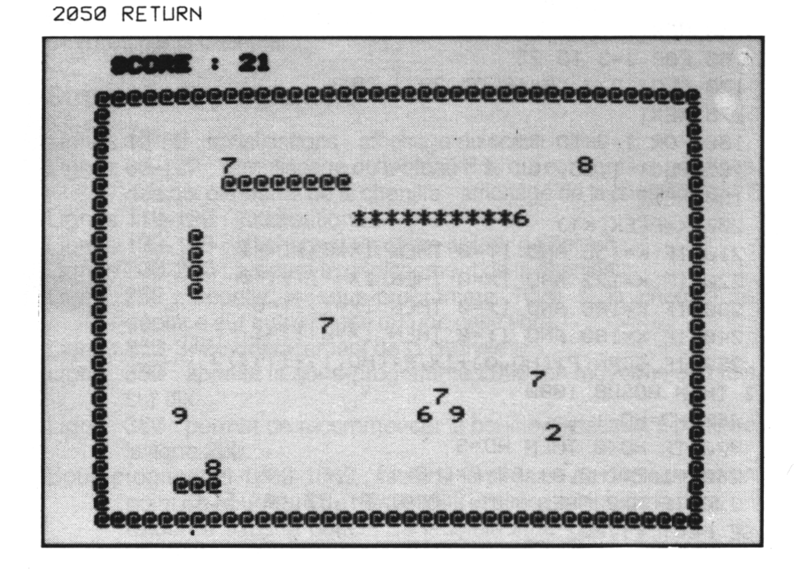
Lignes 1Ø-5Ø : initialisations - affichage du score initial.
Lignes 6Ø-1ØØ : remplissage du tableau P% qui contient les positions des dix éléments de la chenille - affichage de la chenille.
Lignes 11Ø-12Ø : initialisations.
Lignes 13Ø-19Ø : affichage d'un cadre autour de l'écran.
Lignes 2ØØ-24Ø : prépare le déplacement de la chenille.
Ligne 25Ø : appelle le sous-programme 1ØØØ si la chenille se déplace sur autre chose qu'une case libre.
Lignes 26Ø-34Ø : déplacement de la chenille.
Ligne 35Ø : appelle le sous-programme 2ØØØ, en moyenne une fois sur dix.
Ligne 36Ø : permet de recommencer la boucle ci-dessus, à partir de la ligne 2ØØ.
Sous-programme 1ØØØ-1Ø6Ø : teste si la chenille se déplace sur un point de nourriture ou non ; si oui, le score est incrémenté et on retourne dans la boucle de jeu ; sinon le joueur a perdu, on affiche le score et le programme se termine.

Sous-programme 2000-2050 : affiche aléatoirement un point de nourriture sur une case de l'écran, à condition que cette case soit libre.

Programme

```
 10 REM  CHENILLE
 20 DIM P%(9,1)
 30 CLS:PRINT CHR$(20);CHR$(17);CHR$(6):
INK 2:PAPER 0
 40 PLOT 1,0,7
 50 PLOT 3,0,"SCORE  : 0"
 60 FOR I=0 TO 9
 70 P%(I,0)=20
 80 P%(I,1)=12-I
 90 PLOT P%(I,0),P%(I,1),"*"
 100 NEXT
 110 KY=#0208
 120 IX=0:IY=1
 130 FOR I=2 TO 39
 140 PLOT I,2,"@"
 150 NEXT
 160 FOR J=3 TO 25
 170 PLOT 2,J,"@":PLOT 39,J,"@"
 175 NEXT
 180 FOR I=2 TO 39
 185 PLOT I,26,"@"
 190 NEXT
 200 K=PEEK(KY)
 210 IF K=156 AND IY=0 THEN IX=0:IY=-1
 220 IF K=172 AND IX=0 THEN IX=-1:IY=0
 230 IF K=188 AND IX=0 THEN IX=1:IY=0
 240 IF K=180 AND IY=0 THEN IX=0:IY=1
 250 IF SCRN(P%(HD,0)+IX,P%(HD,1)+IY)<>3
2 THEN GOSUB 1000
 260 HD=HD-1
 270 IF HD<0 THEN HD=9
 280 P1=P%(HD,0):P2=P%(HD,1)
 290 IF M>0 THEN M=M-1:PLOT P1,P2,"@" EL
SE PLOT P1,P2," "
```

```
300 HD=HD+1:IF HD=10 THEN HD=0
310 P1=P%(HD,0):P2=P%(HD,1)
320 HD=HD-1:IF HD<0 THEN HD=9
330 P%(HD,0)=P1+IX:P%(HD,1)=P2+IY
340 PLOT P1+IX,P2+IY,"*"
350 IF RND(1)>0.9 THEN GOSUB 2000
360 GOTO 200
1000 CH=SCRN(P%(HD,0)+IX,P%(HD,1)+IY)
1010 IF CH<>64 THEN M=M+CH-48:S=S+CH-48
:PLOT 10,0,STR$(S):PING:RETURN
1020 ZAP:EXPLODE:WAIT 100
1030 CLS
1040 PRINT @ 3,12;"VOTRE SCORE :";S
1050 PRINT CHR$(20);CHR$(17);CHR$(6)
1060 END
2000 C1=INT(RND(1)*36+3)
2010 C2=INT(RND(1)*23+3)
2020 IF SCRN(C1,C2)<>32 THEN RETURN
2030 C3=INT(RND(1)*9+1)
2040 PLOT C1,C2,STR$(C3)
2050 RETURN
```



CHIFFRES ROMAINS

Ce programme convertit en chiffres romains un nombre écrit en chiffres arabes.

Utilisation

Il suffit d'introduire le nombre à convertir, qui doit être entier et positif.

Structure du programme

Lignes 1Ø-4Ø : initialisations - affichage.
Lignes 5Ø-17Ø : cette boucle permet de convertir autant de nombres que l'utilisateur le désire, jusqu'à ce qu'il appuie sur la touche « N » - elle se décompose ainsi :
Lignes 6Ø-7Ø : initialisations - RO$ contiendra l'écriture en chiffres romains du nombre.
Lignes 8Ø-9Ø : introduction du nombre à convertir - sauvegarde de ce nombre N dans la variable N1.
Lignes 1ØØ-13Ø : cette boucle effectue la conversion proprement dite.
Ligne 14Ø : affichage du résultat.
Lignes 15Ø-16Ø : demande pour un autre calcul.
Lignes 21Ø-22Ø : données utiles au programme.

Programme

```
 10 REM  CONVERSION ARABES -> ROMAINS
 20 PAPER 0:INK 2
 30 CLS
 40 PRINT:PRINT " Conversion chiffres ar
abes -> romains"
 50 REPEAT
 60 RO$=""
```

```
 70 RESTORE
 80 PRINT:INPUT "Nombre a convertir ";N
 90 N1=N
 100 REPEAT
 110 READ A,A$
 120 IF N>=A THEN N=N-A:RO$=RO$+A$:GOTO
120
 130 UNTIL N<1
 140 PRINT:PRINT N1;"s'ecrit ";RO$;" en
chiffres romains"
 150 PRINT:PRINT "Un autre nombre a conv
ertir ?"
 160 GET K$
 170 UNTIL K$="N"
 180 DATA 1000,M,900,CM,500,D,400,CD,100
,C,90,XC,50,L,40,XL,10,X,9,IX
 190 DATA 5,V,4,IV,1,I
```

Exemple

```
 Conversion chiffres arabes -> romains
Nombre a convertir ? 42
 42 s'ecrit XLII en chiffres romains
Un autre nombre a convertir ?
Nombre a convertir ? 77
 77 s'ecrit LXXVII en chiffres romains
Nombre a convertir ? 1238
 1238 s'ecrit MCCXXXVIII en chiffres romains
Un autre nombre a convertir ?
Nombre a convertir ? 1984
 1984 s'ecrit MCMLXXXIV en chiffres romains
Un autre nombre a convertir ?
```

MESSAGE



Ce programme vous permet de faire défiler sur l'écran un message de votre choix.

Utilisation

Il suffit d'introduire le message, qui défile immédiatement, en double hauteur.

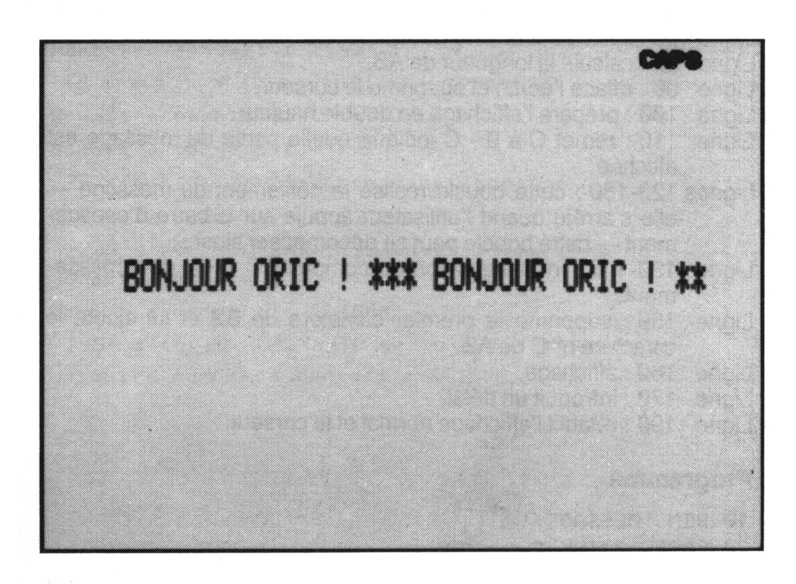
Structure du programme

Lignes 1Ø-4Ø : initialisations - VI contrôle la vitesse de défilement (si VI augmente la vitesse diminue, et réciproquement).
Ligne 5Ø : introduction du message.
Ligne 6Ø : rajoute trois étoiles devant le message.
Ligne 7Ø : initialise la chaîne B$, qui sera affichée.
Ligne 8Ø : calcule la longueur de A$.
Ligne 9Ø : efface l'écran et supprime le curseur.
Ligne 1ØØ : prépare l'affichage en double hauteur.
Ligne 11Ø : remet C à Ø - C indique quelle partie du message est affichée.
Lignes 12Ø-18Ø : cette boucle réalise le défilement du message — elle s'arrête quand l'utilisateur appuie sur la barre d'espacement — cette boucle peut se décomposer ainsi :
Lignes 13Ø-14Ø : incrémentation du compteur C - test de dépassement.
Ligne 15Ø : supprime le premier caractère de B$ et lui ajoute le caractère n° C de A$.
Ligne 16Ø : affichage.
Ligne 17Ø : introduit un délai.
Ligne 19Ø : rétablit l'affichage normal et le curseur.

Programme

```
10 REM  MESSAGE
20 PAPER 0:INK 2
```

```
30 CLS
40 VI=20
50 PRINT:INPUT " Quel est votre message
";A$
60 A$=" *** "+A$
70 B$="
    "
80 L=LEN(A$)
90 CLS:PRINT CHR$(17)
100 PRINT @ 2,11;CHR$(4);CHR$(27);"J"
110 C=0
120 REPEAT
130 C=C+1
140 IF C>L THEN C=1
150 B$=RIGHT$(B$+MID$(A$,C,1),37)
160 PRINT @ 3,11;B$
170 WAIT VI
180 UNTIL KEY$=" "
190 PRINT CHR$(4);CHR$(17)
```



NOMBRE DE JOURS ENTRE DEUX DATES

Ce programme calcule le nombre de jours entre deux dates données, en tenant compte des années bissextiles.

Utilisation

On doit indiquer les deux dates de la façon suivante :

- jour, de 1 à 28, 29, 3Ø ou 31 ;
- mois, de 1 à 12 ;
- année, en entier (ex. : 1984).

Structure du programme

Lignes 1Ø-4Ø : initialisations - affichage.
Lignes 5Ø-8Ø : introduction de la première date.
Lignes 9Ø-1ØØ : début du calcul.
Lignes 11Ø-14Ø : introduction de la seconde date.
Lignes 15Ø-17Ø : suite du calcul - affichage du résultat.
Lignes 18Ø-21Ø : demande pour un autre calcul.
Sous-programme 1ØØØ-1Ø3Ø : permet le calcul du nombre de jours.

Programme

```
 1Ø REM  NOMBRE DE JOURS ENTRE DEUX DATE
S
 15 PAPER Ø:INK 2
 2Ø CLS:PRINT " Ce programme calcule le
nombre de"
 3Ø PRINT "jours entre deux dates, en te
nant"
```

```
 40 PRINT "compte des annees bissextiles
 "
 50 PRINT:PRINT:PRINT "1ere date :"
 60 INPUT "Jour ";J
 70 INPUT "Mois ";M
 80 INPUT "Annee ";A
 90 GOSUB 1000
 100 N1=N
 110 PRINT:PRINT:PRINT "2eme date :"
 120 INPUT "Jour ";J
 130 INPUT "Mois ";M
 140 INPUT "Annee ";A
 150 GOSUB 1000
 160 N2=N
 170 PRINT:PRINT:PRINT " Il y a";ABS(N2-
N1);"jours entre ces deux dates"
 180 PRINT:PRINT:PRINT " Voulez-vous fai
re un autre calcul ?"
 190 GET K$
 200 IF K$<>"N" GOTO 50
 210 END
 1000 N=365*A+31*(M-1)+J
 1010 IF M<3 THEN N=N+INT((A-1)/4)-INT((
A-1)/100)+INT((A-1)/400):RETURN
 1020 N=N+INT(A/4)-INT((M-1)*0.4+2.7)-IN
T(A/100)+INT(A/400)
 1030 RETURN
```

Exemple

Ce programme calcule le nombre de jours entre deux dates, en tenant compte des annees bissextiles.

```
1ere date :
Jour  15
Mois  4
Annee  1984
```

```
2eme date :
Jour  20
Mois  5
Annee  1963

 Il y a 7636 jours entre ces deux dates

 Voulez-vous faire un autre calcul ?
```

CALENDRIER

Vous souvenez-vous si vous êtes né un lundi, un mardi ou un autre jour...? Voulez-vous posséder avant tout le monde le calendrier de l'an 2ØØØ? Tout cela est possible grâce à ce programme qui édite le calendrier de l'année de votre choix.

Utilisation

Il faut introduire l'année (par ex. : 1984, et non simplement 84, car le programme marche pour n'importe quelle année postérieure à 1582). Puis il faut choisir si l'on veut afficher le calendrier sur l'écran ou sur l'imprimante.

Structure du programme

Lignes 1Ø-5Ø : initialisations - affichage.
Ligne 6Ø : introduction de l'année.
Lignes 7Ø-1ØØ : choix de l'affichage.
Lignes 12Ø-13Ø : calcul du premier jour de l'année, indiqué par J (Ø=dimanche... 7=samedi).
Ligne 14Ø : calcul de la bissextilité de l'année (B=−1 pour une année bissextile).
Lignes 15Ø-165 : début de l'affichage.
Lignes 17Ø-27Ø : cette boucle réalise l'affichage du calendrier pour douze mois consécutifs — on peut la décomposer comme ceci :
Lignes 18Ø-2ØØ : lecture du nom du mois M$, du nombre de jours dans ce mois NB - affichage.
Lignes 21Ø-25Ø : cette boucle affiche les numéros des jours du mois (de 1 à 28, 29, 3Ø ou 31 selon le cas) sous les noms correspondants des jours.
Lignes 26Ø-265 : saute une ligne entre deux mois.
Lignes 1ØØØ-1Ø1Ø : données utiles au programme.

Programme

```
 10 REM  CALENDRIER
 20 PAPER 0:INK 2
 30 CLS
 40 PRINT " Ce programme vous donne le c
alendrier";
 50 PRINT "de l'annee de votre choix."
 60 PRINT:INPUT " Annee ";A
 70 PRINT:PRINT " Affichage sur :"
 80 PRINT:PRINT "  1  Ecran"
 90 PRINT:PRINT "  2  Imprimante"
 100 GET K$
 110 IF K$<"1" OR K$>"2" GOTO 100
 120 B=A-1:N=365*A+INT(B/4)-INT(B/100)+I
NT(B/400)-724640
 130 J=N-7*INT(N/7)
 140 B=(A=4*INT(A/4))AND((A<>100*INT(A/1
00))OR(A=400*INT(A/400)))
 150 CLS:J$=" Dim Lun Mar Mer Jeu Ven Sa
m"
 160 IF K$="1" THEN PRINT A:PRINT
 165 IF K$="2" THEN LPRINT A:LPRINT
 170 FOR I=1 TO 12
 180 READ M$,NB
 190 IF K$="1" THEN PRINT TAB(14) M$:PRI
NT
 195 IF K$="2" THEN LPRINT TAB(12) M$:LP
RINT
 200 IF K$="1" THEN PRINT J$ ELSE LPRINT
 J$
 210 FOR N=1 TO NB-((I=2)AND B)
 220 IF K$="1" THEN PRINT TAB(4*J+2);N;
ELSE LPRINT TAB(4*J);N;
 230 J=J+1
 240 IF J=7 THEN J=0:IF K$="1" THEN PRIN
T ELSE LPRINT
 250 NEXT N
 260 IF K$="1" THEN PRINT:PRINT
 265 IF K$="2" THEN LPRINT:LPRINT
```

```
 270 NEXT I
 1000 DATA JANVIER,31,FEVRIER,28,MARS,31
,AVRIL,30,MAI,31,JUIN,30
 1010 DATA JUILLET,31,AOUT,31,SEPTEMBRE,
30,OCTOBRE,31,NOVEMBRE,30,DECEMBRE,31
```

Exemple

Ce programme vous donne le calendrier de l'annee de votre choix.

```
 Annee ? 1985

 Affichage sur :

   1  Ecran
   2  Imprimante

1985

              JANVIER

Dim Lun Mar Mer Jeu Ven Sam
         1   2   3   4   5
6   7   8   9   10  11  12
13  14  15  16  17  18  19
20  21  22  23  24  25  26
27  28  29  30  31

              FEVRIER

Dim Lun Mar Mer Jeu Ven Sam
                     1   2
3   4   5   6   7   8   9
10  11  12  13  14  15  16
17  18  19  20  21  22  23
24  25  26  27  28
```

MARS

Dim	Lun	Mar	Mer	Jeu	Ven	Sam
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

CALCULATRICE



Ce programme, qui simule une calculatrice travaillant en notation polonaise inverse, est intéressant à plusieurs titres. Il vous permettra de vous familiariser avec la notation polonaise inverse si vous ne la connaissiez pas, et d'effectuer aisément certaines opérations mathématiques dont ne dispose pas le Basic de l'Oric (particulièrement les fonctions trigonométriques inverses et trigonométriques hyperboliques). Enfin, il vous montre qu'à partir d'une machine ayant un langage donné (ici le Basic de l'Oric) on peut très bien simuler le fonctionnement d'une machine ayant un autre langage.

Utilisation

En haut de l'écran sont affichés les quatre registres de la pile, X, Y, Z et T. Le registre X est mis en valeur car lui seul est normalement affiché sur une calculatrice et contient le résultat des opérations. Un point d'interrogation apparaît en bas de l'écran. Il suffit alors d'introduire un nombre ou bien un code d'opération, suivi de « Return ». Par exemple, pour diviser 7 par 2, on fera : 7 (Return), 2 (Return), /(Return) ; le registre X affiche alors 3,5.

Notre calculatrice connaît les opérations suivantes : addition (+), soustraction (−), multiplication (*), division (/), élévation à la puissance (Y↑X), élévation au carré (X↑2), racine carrée (SQRX), inversion (1/X), logarithmes naturel (LN) et décimal (LOG), exponentiations de bases e (EXP) et 10 (10↑X), conversions de degrés en radians (>RAD) et de radians en degrés (>DEG), fonctions trigonométriques circulaires (SIN, COS, TAN, calculées en radians) et leurs fonctions inverses (ASN, ACS, ATN, résultat en radians), fonctions trigonométriques hyperboliques (SNH, CSH, TNH) et leurs fonctions inverses (ASH, ACH, ATH), changement de signe (CHS), valeur absolue (ABS), partie entière (INT), partie fractionnaire (FRAC), calcul de pourcentages (% : X reçoit la valeur de Y * X/1ØØ), introduction du nombre π (PI), remise à Ø du registre X (CLX) ou des quatre registres (CLST),

échange des registres X et Y (X<>Y), rotation de la pile vers le bas (R!, c'est-à-dire : X←Y←Z←T ↰).

Structure du programme

Lignes 1Ø-3Ø : initialisations.
Lignes 4Ø-8Ø : affichage des registres.
Lignes 9Ø-1ØØ : introduction d'un nombre ou d'un code d'opération.
Ligne 11Ø : remise à Ø de la variable OP qui indique quelle opération on devra effectuer.
Lignes 12Ø-47Ø : teste successivement les codes d'opération pour mettre OP à la bonne valeur.
Lignes 48Ø-81Ø : choisit le bon sous-programme, selon la valeur de OP, et revient en 3Ø pour un nouveau cycle.
Sous-programmes 1ØØØ, 11ØØ, 12ØØ,... 46ØØ : ils effectuent les opérations que nous avons décrites.

Programme

```
10 REM  CALCULATRICE
20 PAPER 0
30 INK 2
40 CLS
50 PRINT "T =";T
60 PRINT "Z =";Z
70 PRINT "Y =";Y
80 PRINT:PRINT "X =";X
90 PRINT @ 2,25;"";
100 INPUT R$
110 OP=0
120 IF R$="+" THEN OP=1
130 IF R$="-" THEN OP=2
140 IF R$="*" THEN OP=3
150 IF R$="/" THEN OP=4
160 IF R$="CHS" THEN OP=5
170 IF R$="CLX" THEN OP=6
180 IF R$="CLST" THEN OP=7
190 IF R$="INT" THEN OP=8
200 IF R$="FRAC" THEN OP=9
210 IF R$="ABS" THEN OP=10
```

```
220 IF R$="X<>Y" THEN OP=11
230 IF R$="R↓" THEN OP=12
240 IF R$="%" THEN OP=13
250 IF R$=">RAD" THEN OP=14
260 IF R$=">DEG" THEN OP=15
270 IF R$="SIN" THEN OP=16
280 IF R$="COS" THEN OP=17
290 IF R$="TAN" THEN OP=18
300 IF R$="ASN" THEN OP=19
310 IF R$="ACS" THEN OP=20
320 IF R$="ATN" THEN OP=21
330 IF R$="LN" THEN OP=22
340 IF R$="LOG" THEN OP=23
350 IF R$="EXP" THEN OP=24
360 IF R$="10^X" THEN OP=25
370 IF R$="SQRX" THEN OP=26
380 IF R$="X^2" THEN OP=27
390 IF R$="Y^X" THEN OP=28
400 IF R$="1/X" THEN OP=29
410 IF R$="PI" THEN OP=30
420 IF R$="SNH" THEN OP=31
430 IF R$="CSH" THEN OP=32
440 IF R$="TNH" THEN OP=33
450 IF R$="ASH" THEN OP=34
460 IF R$="ACH" THEN OP=35
470 IF R$="ATH" THEN OP=36
480 ON INT(OP/10)+1 GOSUB 500,600,700,8
00
490 GOTO 30
500 ON OP+1 GOSUB 1000,1100,1200,1300,1
400,1500,1600,1700,1800,1900
510 RETURN
600 ON OP-9 GOSUB 2000,2100,2200,2300,2
400,2500,2600,2700,2800,2900
610 RETURN
700 ON OP-19 GOSUB 3000,3100,3200,3300,
3400,3500,3600,3700,3800,3900
710 RETURN
800 ON OP-29 GOSUB 4000,4100,4200,4300,
4400,4500,4600
```

```
810 RETURN
1000 T=Z:Z=Y:Y=X:X=VAL(R$)
1010 RETURN
1100 X=Y+X:Y=Z:Z=T
1110 RETURN
1200 X=Y-X:Y=Z:Z=T
1210 RETURN
1300 X=Y*X:Y=Z:Z=T
1310 RETURN
1400 X=Y/X:Y=Z:Z=T
1410 RETURN
1500 X=-X
1510 RETURN
1600 X=0
1610 RETURN
1700 X=0:Y=0:Z=0:T=0
1710 RETURN
1800 X=INT(X)
1810 RETURN
1900 X=X-INT(X)
1910 RETURN
2000 X=ABS(X)
2010 RETURN
2100 I=X:X=Y:Y=I
2110 RETURN
2200 I=X:X=Y:Y=Z:Z=T:T=I
2210 RETURN
2300 X=X*Y/100
2310 RETURN
2400 X=X*PI/180
2410 RETURN
2500 X=X*180/PI
2510 RETURN
2600 X=SIN(X)
2610 RETURN
2700 X=COS(X)
2710 RETURN
2800 X=TAN(X)
2810 RETURN
```

```
 2900 IF ABS(X)=1 THEN X=PI/2*SGN(X) ELS
E X=ATN(X/SQR(1-X*X))
 2910 RETURN
 3000 IF ABS(X)=1 THEN X=PI/2*(1-SGN(X))
 ELSE X=PI/2-ATN(X/SQR(1-X*X))
 3010 RETURN
 3100 X=ATN(X)
 3110 RETURN
 3200 X=LN(X)
 3210 RETURN
 3300 X=LOG(X)
 3310 RETURN
 3400 X=EXP(X)
 3410 RETURN
 3500 X=10^X
 3510 RETURN
 3600 X=SQR(X)
 3610 RETURN
 3700 X=X*X
 3710 RETURN
 3800 X=Y^X:Y=Z:Z=T
 3810 RETURN
 3900 X=1/X
 3910 RETURN
 4000 T=Z:Z=Y:Y=X:X=PI
 4010 RETURN
 4100 X=(EXP(X)-EXP(-X))/2
 4110 RETURN
 4200 X=(EXP(X)+EXP(-X))/2
 4210 RETURN
 4300 X=(EXP(X)-EXP(-X))/(EXP(X)+EXP(-X)
)
 4310 RETURN
 4400 X=LN(X+SQR(X*X+1))
 4410 RETURN
 4500 X=LN(X+SQR(X*X-1))
 4510 RETURN
 4600 X=LN((1+X)/(1-X))/2
 4610 RETURN
```

NOMBRES PREMIERS

Ce programme recherche tous les nombres premiers jusqu'à une limite fixée.

Utilisation

Il suffit d'indiquer la valeur de cette limite.

Structure du programme

Lignes 1Ø-6Ø : initialisations - affichage.
Lignes 7Ø-1ØØ : introduction de la limite.
Lignes 11Ø-2ØØ : recherche et affichage des nombres premiers.

Programme

```
 10 REM  NOMBRES PREMIERS
 20 PAPER 0:INK 2
 30 CLS
 40 PRINT:PRINT " Ce programme permet d'
afficher la"
 50 PRINT "suite des nombres premiers, j
usqu'a"
 60 PRINT "une limite que vous devez pre
ciser."
 70 PRINT " Jusqu'a quel nombre dois-je"
 80 INPUT "rechercher ";M
 90 M=INT(ABS(M))
 100 IF M<5 GOTO 70
 110 PRINT:PRINT 2,3,
 120 FOR N=5 TO M STEP 2
 130 R=INT(SQR(N))+1
```

```
140 FOR I=3 TO R STEP 2
150 Q=N/I
160 IF Q=INT(Q) THEN I=R:P=0:GOTO 180
170 P=1
180 NEXT I
190 IF P=1 THEN PRINT N,
200 NEXT N
```

Exemple

```
  Ce programme permet d'afficher la
suite des nombres premiers, jusqu'a
une limite que vous devez preciser.
 Jusqu'a quel nombre dois-je
rechercher ? 100

2         3         5         7         11
13        17        19        23        29
31        37        41        43        47
53        59        61        67        71
73        79        83        89        97
```

DECOMPOSITION EN FACTEURS PREMIERS

Ce programme effectue la décomposition en facteurs premiers d'un nombre entier positif, et affiche la liste de tous ses diviseurs premiers.

Utilisation

Introduisez le nombre à étudier, qui doit être un entier supérieur ou égal à 2.

Structure du programme

Lignes 1Ø-7Ø : initialisations - affichage.
Lignes 8Ø-1ØØ : introduction du nombre.
Lignes 11Ø-17Ø : décomposition en facteurs premiers - affichage des diviseurs premiers.

Programme

```
 10 REM  DECOMPOSITION
 20 PAPER 0:INK 2
 30 CLS
 40 PRINT " Ce programme affiche la list
e de tous";
 50 PRINT "les diviseurs premiers d'un n
ombre"
 60 PRINT "entier superieur ou egal a 2.
"
 70 PRINT
 80 INPUT"Votre nombre ";N:PRINT
 90 N=INT(ABS(N))
```

```
100 IF N<2 GOTO 80
110 D=2
120 IF D>SQR(N) THEN PRINT N:GOTO 70
130 Q=N/D
140 IF Q<>INT(Q) THEN D=D+1:GOTO 120
150 PRINT D,
160 N=Q
170 GOTO 120
```

Exemple

Ce programme affiche la liste de tous les diviseurs premiers d'un nombre entier superieur ou egal a 2.

```
Votre nombre ? 184677570

2         3         3         3         3
3         3         5         7         7
11        47

Votre nombre ? 101010101

41        271       9091
```

FACTORIELLES

Ce programme calcule la factorielle d'un nombre entier N positif ou nul. Le résultat est noté N!.

Rappelons que Ø!=1 et (N+1)!=(N+1)×N!

Donc, pour N>O : N!=N×(N−1)×(N−2)×...×3×2×1.

Utilisation

Il suffit d'introduire le nombre dont on veut calculer la factorielle.

Structure du programme

Lignes 1Ø-7Ø : initialisations - affichage.
Lignes 8Ø-9Ø : introduction du nombre.
Ligne 1ØØ : teste si N>33, auquel cas on effectue un calcul approché de N!.
Lignes 11Ø-17Ø : calcul normal de N! par multiplications successives.
Lignes 18Ø-23Ø : calcul approché de N! par la formule de Stirling : pour N assez grand, $N! \approx \sqrt{2\pi N}\, N^N e^{-N}$; le résultat est affiché sous la forme mantisse-exposant, et si l'exposant est très grand, il est lui-même affiché sous la forme mantisse-exposant.

Programme

```
 10 REM  FACTORIELLES
 20 PAPER 0:INK 2
 30 CLS
 40 PRINT " Ce programme vous permet de 
calculer"
 50 PRINT "la factorielle de n'importe q
uel"
```

```
 60 PRINT "nombre entier positif ou nul.
"
 70 PRINT
 80 INPUT "Votre nombre ";N:PRINT
 90 N=INT(ABS(N))
 100 IF N>33 GOTO 180
 110 F=1
 120 IF N=0 GOTO 160
 130 FOR I=1 TO N
 140 F=F*I
 150 NEXT
 160 PRINT N;"! =";F
 170 GOTO 70
 180 F1=(LN(2*PI)/2+LN(N)/2+N*LN(N)-N)/L
N(10)
 190 E=INT(F1)
 200 M=10^(F1-E)
 210 F$=STR$(M)+"E+"+RIGHT$(STR$(E),LEN(
STR$(E))-1)
 220 PRINT N;"! =";F$
 230 GOTO 70
```

Exemple

```
 Ce programme vous permet de calculer
la factorielle de n'importe quel
nombre entier positif ou nul.

Votre nombre ? 32

32 ! = 2.63130837E+35

Votre nombre ? 1984

1984 ! = 5.3737569E+5682

Votre nombre ? 536

536 ! = 6.51864734E+1231
```

```
Votre nombre ? 397432

397432 ! = 7.95043013E+2052726

Votre nombre ? 2.98434577E+10

2.98434577E+10 ! = 1E+2.99644869E+11
```

CHANGEMENT DE BASE

Ce programme effectue la conversion d'un nombre donné d'une base vers une autre.

Utilisation

On doit introduire la base de départ, la base d'arrivée et le nombre à convertir.

Structure du programme

Lignes 1Ø-1ØØ : initialisations - affichage.
Lignes 11Ø-13Ø : introduction des données.
Lignes 14Ø-22Ø : conversion du nombre, de sa base de départ vers la base 1Ø.
Lignes 23Ø-28Ø : conversion du nombre en base 1Ø vers la base d'arrivée.
Lignes 29Ø-3ØØ : affichage du résulat - retour pour un nouveau calcul.

Programme

```
 10 REM  CHANGEMENT DE BASE
 20 PAPER 0:INK 2
 30 CLS
 40 PRINT " Ce programme convertit un no
mbre"
 50 PRINT "entier d'une base quelconque
vers une"
 60 PRINT "autre."
 70 PRINT " Au cas ou la base est superi
eure a 10";
```

```
 80 PRINT "on utilise les lettres A,B,..
.,Z pour"
 90 PRINT "representer les nombres 10,11
,...,36."
 100 PRINT:PRINT
 110 INPUT "Base de depart ";A
 120 INPUT "Base d'arrivee ";B
 130 INPUT "Votre nombre   ";X$
 140 L=LEN(X$)
 150 N=0
 160 FOR I=0 TO L-1
 170 P1=ASC(MID$(X$,L-I,1))
 180 IF P1>47 AND P1<58 THEN P=P1-48:GOT
O 210
 190 IF P1>64 AND P1<91 THEN P=P1-55:GOT
O 210
 200 PRINT "Vous avez fait une erreur.":
GOTO 130
 210 N=N+P*A^I
 220 NEXT
 230 R$=""
 240 Q=INT(N/B)
 250 R=N-B*Q
 260 IF R<10 THEN R$=CHR$(R+48)+R$:GOTO
280
 270 R$=CHR$(R+55)+R$
 280 IF Q>0 THEN N=Q:GOTO 240
 290 PRINT "Le resultat est: ";R$
 300 GOTO 100
```

Exemple

Ce programme convertit un nombre entier d'une base quelconque vers une autre.

Au cas ou la base est superieure a 10 on utilise les lettres A,B,...,Z pour representer les nombres 10,11,...,36.

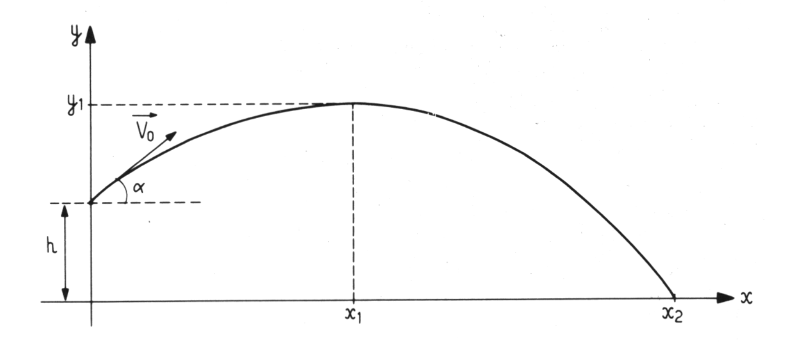
```
Base de depart ? 10
Base d'arrivee ? 16
Votre nombre   ? 65535
Le resultat est: FFFF

Base de depart ? 2
Base d'arrivee ? 16
Votre nombre   ? 11101010
Le resultat est: EA

Base de depart ? 5
Base d'arrivee ? 23
Votre nombre   ? 42231423
Le resultat est: 15LE6
```

BALISTIQUE

Ce programme calcule certains éléments de la trajectoire d'un mobile en chute libre :



La trajectoire est une parabole ; h est la hauteur de départ du mobile, V_0 sa vitesse initiale et α l'angle que fait V_0 avec l'horizontale ; y_1 est la hauteur maximale atteinte par le mobile, x_1 son abscisse à ce moment ; x_2 est l'abscisse du mobile quand il atteint le sol.

Utilisation

On doit indiquer les valeurs de V_0, α , h.

Le programme affiche alors les valeurs de x_1, y_1, t_1 (instant correspondant), puis x_2 et t_2.

Structure du programme

Lignes 1Ø-7Ø : initialisations - affichage.

Ligne 8Ø : G représente l'intensité de la pesanteur et peut être modifié pour étudier des trajectoires en différents endroits de la planète.
Lignes 9Ø-13Ø : introduction des données.
Lignes 14Ø-29Ø : calculs et affichage des résultats.
Ligne 3ØØ : retour pour un autre calcul.

Programme

```
 10 REM  BALISTIQUE
 20 PAPER 0:INK 2
 30 CLS
 40 PRINT " Ce programme vous permet de
calculer"
 50 PRINT "certains parametres de la tra
jectoire"
 60 PRINT "d'un projectile lance d'une h
auteur et";
 70 PRINT "avec une vitesse quelconques.
"
 80 G=9.81
 90 PRINT:PRINT
 100 INPUT "Vitesse initiale  ";V0
 110 INPUT "Angle (en degres) ";A
 120 INPUT "Hauteur de depart ";H
 130 IF H<0 GOTO 120
 140 A=A*PI/180
 150 T1=V0*SIN(A)/G
 160 PRINT
 170 IF T1<=0 THEN PRINT "Le mobile desc
end tout de suite":GOTO 230
 180 PRINT "Le mobile commence a descend
re a :"
 190 PRINT "T =";T1;"secondes"
 200 PRINT "Ses coordonnees sont alors :
"
 210 PRINT "X =";V0*COS(A)*T1;"metres"
 220 PRINT "Y =";-1/2*G*T1^2+V0*SIN(A)*T
1+H;"metres"
```

```
 230 T2=(V0*SIN(A)+SQR((V0*SIN(A))^2+2*G
*H))/G
 240 PRINT
 250 PRINT "Le mobile atteint le sol a :

 260 PRINT "T =";T2;"secondes"
 270 PRINT "Son abscisse est alors :"
 280 PRINT "X =";V0*COS(A)*T2;"metres"
 290 PRINT:PRINT
 300 GOTO 100
```

Exemple

Ce programme vous permet de calculer certains parametres de la trajectoire d'un projectile lance d'une hauteur et avec une vitesse quelconques.

```
Vitesse initiale  ? 10
Angle (en degres) ? 30
Hauteur de depart ? 20

Le mobile commence a descendre a :
T = .509683996 secondes
Ses coordonnees sont alors :
X = 4.41399288 metres
Y = 21.27421 metres

Le mobile atteint le sol a :
T = 2.59229048 secondes
Son abscisse est alors :
X = 22.4498941 metres

Vitesse initiale  ? 5
Angle (en degres) ? 0
Hauteur de depart ? 330

Le mobile descend tout de suite
```

```
Le mobile atteint le sol a :
T = 8.20233428 secondes
Son abscisse est alors :
X = 41.0116714 metres
```

RESOLUTION D'EQUATION PAR DICHOTOMIE

Ce programme permet de résoudre une équation du type f(x)=Ø par dichotomie.

Utilisation

On doit d'abord programmer la fonction en ligne 12Ø.

On doit ensuite indiquer entre quelles bornes l'ordinateur doit rechercher les solutions. Ces bornes ne doivent pas être quelconques ; il faut impérativement que la fonction soit négative à une borne et positive à l'autre, sans quoi la méthode de dichotomie ne peut pas s'appliquer. De plus, s'il y a plusieurs racines entre les bornes que vous indiquez, l'ordinateur n'en trouvera qu'une.

Structure du programme

Lignes 1Ø-5Ø : initialisations - affichage de commentaires.
Lignes 6Ø-1ØØ : demande si l'on veut programmer la fonction ou résoudre l'équation. Ceci est très utile si l'on a oublié de programmer la fonction avant de lancer le programme.
Ligne 11Ø : liste la ligne 12Ø où est définie la fonction si K$=« 1 ».
Ligne 12Ø : définition de la fonction dont on recherche les zéros.
Lignes 13Ø-16Ø : introduction et test des bornes.
Lignes 17Ø-21Ø : cette boucle effectue la recherche par dichotomie proprement dite - elle s'arrête quand on a effectivement trouvé une solution (FNF(C)=Ø) ou quand on a atteint une précision suffisante ($B-A<1Ø^{-9}$).
Lignes 22Ø-26Ø : affichage de la solution - demande pour un nouveau calcul.

Programme

```
 10 REM  RESOLUTION PAR DICHOTOMIE
 20 PAPER 0:INK 2
 30 CLS
 40 PRINT:PRINT " Ce programme resout l'
equation f(x)=0";
 50 PRINT "par la methode de dichotomie.
"
 60 PRINT:PRINT " Voulez-vous :"
 70 PRINT:PRINT "  1  Programmer la fonc
tion"
 80 PRINT:PRINT "  2  Resoudre l'equatio
n"
 90 GET K$
 100 IF K$<"1" OR K$>"2" GOTO 90
 110 IF K$="1" THEN CLS:PRINT "Allez-y :
":LIST 120
 120 DEF FN F(X)=COS(X)-X
 130 PRINT:PRINT
 140 INPUT "Borne inferieure ";A
 150 INPUT "Borne superieure ";B
 160 IF FNF(A)*FNF(B)>=0 OR B<A THEN PRI
NT:PRINT "Bornes incorrectes":GOTO 130
 170 REPEAT
 180 C=(A+B)/2
 190 IF FN F(A)*FN F(C)>0 THEN A=C
 200 IF FN F(B)*FN F(C)>0 THEN B=C
 210 UNTIL FN F(C)=0 OR B-A<1E-9
 220 PRINT:PRINT
 230 PRINT "La solution est X=";C
 240 PRINT:PRINT "Voulez-vous faire un a
utre calcul ?"
 250 GET K$
 260 IF K$<>"N" GOTO 30
```

Exemple

```
  Ce programme resout l'equation f(x)=0
par la methode de dichotomie.
```

```
 Voulez-vous :

   1  Programmer la fonction
   2  Resoudre l'equation

120 DEF FN F(X)=COS(X)-X


 Voulez-vous :

   1  Programmer la fonction
   2  Resoudre l'equation

Borne inferieure ? 0
Borne superieure ? 1.5

La solution est X= .739085132

Voulez-vous faire un autre calcul ?
```

INTEGRATION

Ce programme calcule l'intégrale d'une fonction entre deux bornes finies par la méthode de Simpson.

Utilisation

Il faut d'abord programmer la fonction en ligne 200.

Après avoir relancé le programme, on doit indiquer les bornes d'intégration et le pas d'intégration.

Structure du programme

Lignes 10-70 : initialisations - affichage.
Lignes 80-140 : permet de revenir à la programmation de la fonction, ou bien de continuer le programme pour calculer l'intégrale.
Ligne 150 : liste la ligne 200, si l'utilisateur l'a demandée.
Ligne 200 : définition de la fonction à intégrer.
Lignes 210-260 : introduction des données.
Lignes 270-360 : calcul de l'intégrale par la méthode de Simpson.
Lignes 370-380 : affichage du résultat.
Lignes 390-430 : demande pour un autre calcul.

Programme

```
 10 REM  INTEGRATION
 20 PAPER 0:INK 2
 30 CLS
 40 PRINT "Ce programme calcule l'integr
ale d'une";
 50 PRINT "fonction entre deux bornes fi
nies par"
 60 PRINT "la methode de Simpson."
```

```
70 PRINT
80 PRINT "Voulez-vous :"
90 PRINT
100 PRINT "   1  Programmer la fonction
"
110 PRINT
120 PRINT "   2  Calculer son integrale
"
130 GET K$
140 IF K$<"1" OR K$>"2" GOTO 130
150 IF K$="1" THEN CLS:PRINT "Allez-y :
":LIST 200
200 DEF FN F(X)=EXP(-X^2)
210 PRINT:PRINT "Bornes d'integration :
"
220 PRINT:INPUT "  inferieure ";A
230 PRINT:INPUT "  superieure ";B
240 PRINT:INPUT "Pas d'integration ";N
250 N=INT(ABS(N))
260 IF N<2 GOTO 240
270 N=2*N
280 H=(B-A)/N
290 I=FN F(A)+FN F(B)
300 FOR K=2 TO N-2 STEP 2
310 I=I+2*FN F(A+K*H)
320 NEXT
330 FOR K=1 TO N-1 STEP 2
340 I=I+4*FN F(A+K*H)
350 NEXT
360 I=I*H/3
370 PRINT:PRINT "Le resultat est :"
380 PRINT:PRINT "  I =";I
390 PRINT:PRINT "Voulez-vous calculer u
ne autre"
400 PRINT "integrale ?"
410 GET K$
420 IF K$="" GOTO 410
430 IF K$<>"N" GOTO 80
```

Exemple

```
Ce programme calcule l'integrale d'une
fonction entre deux bornes finies par
la methode de Simpson.
Voulez-vous :
    1  Programmer la fonction.
    2  Calculer son integrale.

Allez-y :

200 DEF FN F(X)=EXP(-X^2)


Voulez-vous :
    1  Programmer la fonction.
    2  Calculer son integrale.

Bornes d'integration :
   inferieure ?-1
   superieure ? 1
Pas d'integration ? 20

Le resultat est :

   I = 1.49364837

Voulez-vous calculer une autre
integrale ?
```

EQUATION COMPLEXE DU SECOND DEGRE

Ce programme calcule les deux racines d'une équation du second degré à coefficients complexes, de la forme $az^2+bz+c=0$, où a, b, c et z sont donc des nombres complexes.

Utilisation

Il faut introduire la partie réelle, puis la partie imaginaire, de chaque coefficient a, b, c. Le programme affiche alors les résultats.

Structure du programme

Lignes 10-40 : initialisations - affichage.
Lignes 50-130 : introduction des coefficients a, b, c.
Lignes 140-150 : calcul du discriminant de l'équation.
Lignes 160-170 : calcul d'une racine carrée de ce discriminant.
Lignes 180-250 : calcul des deux racines.
Lignes 260-320 : affichage des résultats.
Lignes 330-360 : demande pour un autre calcul.

Programme

```
 10 REM  EQUATION DU SECOND DEGRE
         A COEFFICIENTS COMPLEXES
 20 CLS:PAPER 0:INK 2
 30 PRINT:PRINT "Resolution d'equation d
u type :"
 40 PRINT:PRINT "    A * Z^2 + B * Z + C
= 0"
```

```
 50 PRINT:PRINT:PRINT "Coefficient A :"
 60 INPUT "Partie reelle      ";A0
 70 INPUT "Partie imaginaire ";A1
 80 PRINT:PRINT "Coefficient B :"
 90 INPUT "Partie reelle      ";B0
 100 INPUT "Partie imaginaire ";B1
 110 PRINT:PRINT "Coefficient C :"
 120 INPUT "Partie reelle      ";C0
 130 INPUT "Partie imaginaire ";C1
 140 D0=B0*B0-B1*B1+4*(A1*C1-A0*C0)
 150 D1=2*B0*B1-4*(A0*C1+A1*C0)
 160 E0=SQR((SQR(D0*D0+D1*D1)+D0)/2)
 170 E1=((SGN(D1)<0)*2+1)*SQR((SQR(D0*D0
+D1*D1)-D0)/2)
 180 F0=(E0-B0)/(A0*A0+A1*A1)/2
 190 F1=(E1-B1)/(A0*A0+A1*A1)/2
 200 G0=-(E0+B0)/(A0*A0+A1*A1)/2
 210 G1=-(E1+B1)/(A0*A0+A1*A1)/2
 220 H0=A0*F0+A1*F1
 230 H1=A0*F1-A1*F0
 240 I0=A0*G0+A1*G1
 250 I1=A0*G1-A1*G0
 260 PRINT:PRINT:PRINT "Voici les deux r
acines :"
 270 PRINT:PRINT "Racine Z1 :"
 280 PRINT "Partie reelle      :";H0
 290 PRINT "Partie imaginaire :";H1
 300 PRINT:PRINT "Racine Z2 :"
 310 PRINT "Partie reelle      :";I0
 320 PRINT "Partie imaginaire :";I1
 330 PRINT:PRINT:PRINT "Voulez-vous reso
udre une autre"
 340 PRINT " equation ?"
 350 GET K$
 360 IF K$<>"N" GOTO 20
```

Exemple

```
Resolution d'equation du type :

   A * Z^2 + B * Z + C = 0

Coefficient A :
Partie reelle      ? 1
Partie imaginaire ? 0

Coefficient B :
Partie reelle      ? 0
Partie imaginaire ? 0

Coefficient C :
Partie reelle      ? 1
Partie imaginaire ? 0

Voici les deux racines :

Racine Z1 :
Partie reelle      : 0
Partie imaginaire : 1

Racine Z2 :
Partie reelle      : 0
Partie imaginaire :-1

Voulez-vous resoudre une autre
equation ?
```

COURBES

Ce programme trace en haute résolution le graphe d'une fonction.

Utilisation

Dans une première étape, vous programmez la fonction à représenter en modifiant la ligne 16Ø. Après avoir relancé le programme, vous devez donner les bornes de l'intervalle à étudier. L'ordinateur vous demande alors si vous désirez choisir les bornes sur l'axe des ordonnées. Si vous ne connaissez pas *a priori* les bornes atteintes par F(X), le programme les calcule à votre place. Cependant, si la fonction prend des valeurs très élevées (par ex. F[X]=1/X, au voisinage de Ø), la courbe semblera être écrasée sur les axes. Il faut alors relancer le programme en choisissant convenablement les bornes sur l'axe Y.

Structure du programme

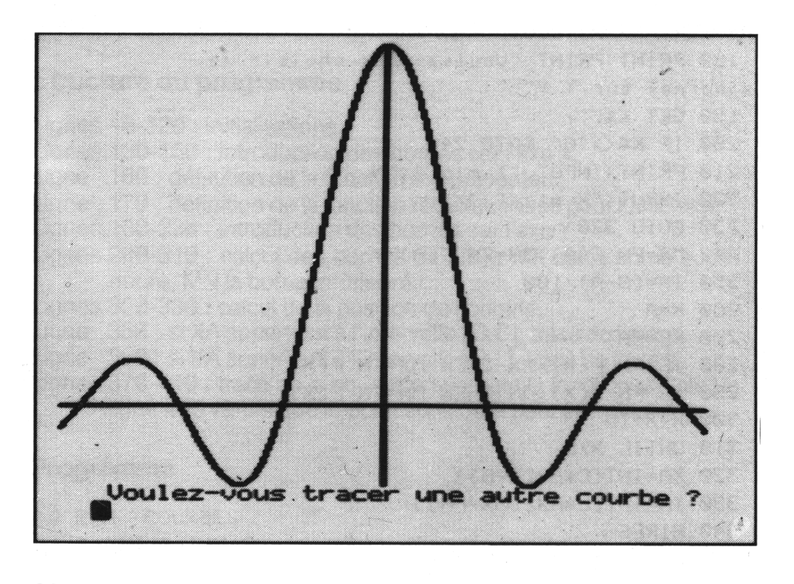
Lignes 1Ø-12Ø : initialisations.
Lignes 13Ø-15Ø : introduction des bornes sur l'axe X.
Ligne 16Ø : définition de la fonction à représenter.
Ligne 17Ø : définition de la fonction réduite utilisée pour l'affichage.
Lignes 18Ø-23Ø : introduction des bornes sur l'axe Y.
Lignes 24Ø-31Ø : calcul des bornes de F(X) : MX est la borne supérieure, MN la borne inférieure.
Lignes 32Ø-33Ø : calcul de la position de l'origine.
Ligne 35Ø : si XA appartient à l'intervalle [O,C], tracé de l'axe Y.
Ligne 36Ø : si YA appartient à l'intervalle [O,L], tracé de l'axe X.
Lignes 37Ø-44Ø : tracé de la courbe en utilisant la fonction réduite ; la ligne 4ØØ vérifie que le point considéré peut être affiché.

Programme

```
10 REM  COURBES
20 PAPER 0:INK 2
```

```
 30 CLS
 40 C=239:L=199
 50 PRINT " Ce programme trace la repres
entation"
 60 PRINT "graphique d'une fonction Y=f(
X)."
 70 PRINT:PRINT " Voulez-vous :"
 80 PRINT:PRINT "  1  Programmer la fonc
tion"
 90 PRINT:PRINT "  2  Tracer la courbe"
 100 GET K$
 110 IF K$<"1" OR K$>"2" GOTO 100
 120 IF K$="1" THEN CLS:PRINT "Allez-y :
":LIST 160
 130 PRINT:PRINT "Introduisez les bornes
 :"
 140 PRINT:INPUT "X min ";A
 150 INPUT "X max ";B
 160 DEF FN F(X)=SIN(X)/X
 170 DEF FN G(X)=INT((FN F(A+X*(B-A)/C)-
MX)*L/(MN-MX)+0.5)
 180 PRINT:PRINT "Voulez-vous choisir le
s bornes sur Y ?"
 190 GET K$
 200 IF K$<>"O" GOTO 240
 210 PRINT:INPUT "Y min ";MN
 220 INPUT "X min ";MX
 230 GOTO 320
 240 MX=FN F(A):MN=FN F(A)
 250 IN=(B-A)/100
 260 X=A
 270 REPEAT
 280 IF FN F(X)>MX THEN MX=FN F(X)
 290 IF FN F(X)<MN THEN MN=FN F(X)
 300 X=X+IN
 310 UNTIL X>B
 320 XA=INT(C*A/(A-B))
 330 YA=INT(L*MX/(MX-MN))
 340 HIRES
```

```
 350 IF XA>=0 AND XA<=C THEN CURSET XA,0
,1:DRAW 0,L,1
 360 IF YA>=0 AND YA<=L THEN CURSET 0,YA
,1:DRAW C,0,1
 370 XP=0
 380 YP=FN G(0)
 390 FOR I=1 TO C
 400 IF YP<0 OR YP>L OR FN G(I)<0 OR FN
G(I)>L GOTO 420
 410 CURSET XP,YP,1:DRAW 1,FN G(I)-YP,1
 420 XP=I
 430 YP=FN G(I)
 440 NEXT I
 450 PRINT " Voulez-vous tracer une autr
e courbe ?"
 460 GET K$:GET K$
 470 IF K$<>"N" THEN CLS:TEXT:GOTO 70
 480 TEXT
```



CALENDRIERS JULIEN ET GREGORIEN

Le calendrier Julien, institué par Jules César en 45 av. J.-C., prenait comme origine le 1er janvier de l'an 4713 av. J.-C. à midi et cumulait le nombre de jours écoulés depuis cette date. Il fut remplacé en 1582 par le calendrier Grégorien que nous connaissons aujourd'hui.

Le programme suivant vous permet, à partir du jour Julien, de calculer la date correspondante dans le calendrier Grégorien.

Utilisation

Il suffit de donner le numéro du jour Julien.

Structure du programme

Lignes 1Ø-3Ø : initialisations.
Lignes 4Ø-5Ø : introduction du jour Julien.
Lignes 6Ø-18Ø : calcul de la date dans le calendrier Grégorien, représentée par les variables JO, M et A.
Lignes 19Ø-21Ø : affichage du résultat.
Lignes 22Ø-25Ø : demande pour un autre calcul.

Programme

```
 10 REM  CALENDRIERS JULIEN ET GREGORIEN
 20 PAPER 0:INK 2
 30 CLS
 40 PRINT:PRINT " Quel est le jour Julie
n envisage ?"
```

```
 50 PRINT:INPUT J:PRINT
 60 J=INT(ABS(J))
 70 IF J<2299161 GOTO 100
 80 X=INT((J-1867216.25)/36524.25)
 90 J=J+1+X-INT(X/4)
 100 Y=J+1524
 110 Z1=INT((Y-122.1)/365.25)
 120 T=INT(365.25*Z1)
 130 Z2=INT((Y-T)/30.6001)
 140 IF Z2<=13.5 THEN M=Z2-1:GOTO 160
 150 M=Z2-13
 160 JO=Y-T-INT(30.6001*Z2)
 170 IF Z2<=2.5 THEN A=Z1-4715:GOTO 190
 180 A=Z1-4716
 190 PRINT " Cette date correspond au :"
 200 PRINT:PRINT JO;"/";M;"/";A
 210 PRINT:PRINT "dans le calendrier Gre
gorien."
 220 PRINT:PRINT:PRINT " Voulez-vous cal
culer une nouvelle"
 230 PRINT "date ?"
 240 GET K$
 250 IF K$<>"N" GOTO 40
```

Exemple

```
 Quel est le jour Julien envisage ?
? 2444000
 Cette date correspond au :
 6 / 5 / 1979
dans le calendrier Gregorien.

 Voulez-vous calculer une nouvelle
date ?
```

PHASES DE LA LUNE

Ce programme permet de calculer l'âge de la lune pour toute date comprise entre 1900 et 2100.

Utilisation

Il suffit de fournir la date considérée, l'ordinateur donne alors un résultat compris entre 0 et 29,5 jours. 0 correspond à la nouvelle lune, 7,4 au premier quartier, 14,8 à la pleine lune, 22,1 au dernier quartier et 29,5 à la nouvelle lune.

Structure du programme

Lignes 10-50 : initialisations.
Lignes 60-70 : introduction de la date.
Lignes 80-110 : calcul de l'âge de la lune, sachant que les phases se succèdent avec une période de 29,5 jours environ.
Lignes 120-130 : affichage du résultat.
Lignes 140-170 : demande pour un autre calcul.
Sous-programme 200-220 : calcule le nombre de jours entre la date et une date de référence qui correspond à une nouvelle lune.

Programme

```
 10 REM  PHASES DE LA LUNE
 20 PAPER 0:INK 2
 30 CLS
 40 PRINT:PRINT "Ce programme vous perme
t de determiner"
 50 PRINT "les phases de la lune."
 60 PRINT:PRINT "Entrez la date sous la
forme :"
```

```
 70 INPUT " JJ,MM,AAAA ";J,M,A
 80 M1=M:A1=A
 90 GOSUB 200
 100 T=29.530589
 110 N=N/T+5000
 120 AG=(N-INT(N))*T
 130 PRINT:PRINT "Le";J;"/";M1;"/";A1;",
 la lune aura"
 140 PRINT AG;"jours."
 150 PRINT:PRINT "Voulez-vous faire un a
utre calcul ?"
 160 GET K$
 170 IF K$<>"N" GOTO 60
 180 END
 200 IF M<=2 THEN A=A-1:M=M+13 ELSE M=M+
1
 210 N=J+INT(365.25*A)+INT(30.6001*M)-72
3067
 220 RETURN
```

Exemple

```
Ce programme vous permet de determiner
les phases de la lune.

Entrez la date sous la forme :
 JJ,MM,AAAA ? 15 ,4 ,1984

Le 15 / 4 / 1984 , la lune aura
14.695226 jours.

Voulez-vous faire un autre calcul ?
```

LEVER ET COUCHER DU SOLEIL

Ce programme détermine les heures de lever et de coucher du soleil, son azimut à ces instants, ainsi que la durée de l'ensoleillement pour la date considérée.

Utilisation

On doit indiquer :

– la date d'observation,
– la longitude et la latitude de l'observateur, exprimées en degrés et fraction de degré.

Les heures de lever et de coucher du soleil sont exprimées en temps universel (TU).

Structure du programme

Lignes 1Ø-5Ø : initialisations - affichage.
Lignes 6Ø-1ØØ : introduction des données.
Lignes 1Ø5-25Ø : calcul des résultats.
Lignes 26Ø-37Ø : affichage des résultats.
Lignes 38Ø-41Ø : demande pour un autre calcul.
Sous-programme 2ØØØ-2Ø5Ø : calcul du nombre de jours écoulés depuis le début de l'année.
Sous-programme 3ØØØ-3Ø3Ø : conversion des heures et fraction d'heure en heures et minutes.

Programme

```
10 REM  SOLEIL
15 PAPER 0:INK 2
```

```
 20 CLS:PRINT " Ce programme calcule l'h
eure du lever";
 30 PRINT "et l'heure du coucher du sole
il,la"
 40 PRINT "duree de l'ensoleillement ain
si que"
 50 PRINT "l'azimut du lever et du couch
er."
 60 PRINT:PRINT:PRINT "Entrez la date d'
observation sous la"
 70 INPUT "forme JJ,MM,AAAA ";J,M,AN
 80 PRINT:PRINT "Entrez la longitude (co
mptee vers"
 90 PRINT "l'est positivement et vers l'
ouest"
 95 INPUT "negativement) ";LO
 100 PRINT:INPUT "Entrez la latitude ";L
T
 105 C0=PI/180
 110 GOSUB 2000
 120 LE=282.63+0.9856*(NJ-3)+1.9167*SIN(
(0.9856*(NJ-3))*C0)
 130 IF LE>360 THEN LE=LE-360
 140 AD=(ATN((TAN(LE*C0))*0.9175))/C0
 143 IF AD<=0 THEN AD=AD+360
 147 AD=AD/15
 150 X=SIN(LE*C0)*0.3978
 160 DE=(ATN(X/SQR(1-X^2)))/C0
 170 X=TAN(LT*C0)*TAN(DE*C0)
 180 EN=((ATN(X/SQR(1-X^2)))/C0+90)*2/15
 190 TP=AD-(6.6224+0.06571*(NJ+0.5)+LO/1
5)
 200 IF TP<=0 THEN TP=TP+24
 210 LU=TP-EN/2
 220 CO=TP+EN/2
 230 X=SIN(DE*C0)/COS(LT*C0)
 240 AL=90-(ATN(X/SQR(1-X^2)))/C0
 250 AC=360-AL
 260 PRINT:PRINT:PRINT "Pour la date con
sideree, en T.U.,"
```

```
270 PRINT "le soleil se levera a :"
280 Y=LV
290 GOSUB 3000
300 PRINT "et se couchera a :"
310 Y=CO
320 GOSUB 3000
330 PRINT "pour un ensoleillement total
de "
340 Y=EN
350 GOSUB 3000
360 PRINT:PRINT "Azimut du lever  :";AL
;"DEG"
370 PRINT "Azimut du coucher:";AC;"DEG"

380 PRINT:PRINT:PRINT "Voulez-vous fair
e un autre calcul ?"
390 GET K$
400 IF K$<>"N" GOTO 60
410 END
2000 IF M<=2 THEN AN=AN-1:M=M+13:GOTO 2
020
2010 M=M+1
2020 D=J+INT(365.25*AN)+INT(30.6001*M)
2030 D0=428+INT(365.25*(AN-1))
2040 NJ=D-D0
2050 RETURN
3000 H=INT(Y)
3010 MN=INT((Y-H)*300/5+0.5)
3020 PRINT H;"H";MN;"MN"
3030 RETURN
```

Exemple

```
 Ce programme calcule l'heure du lever
et l'heure du coucher du soleil,la
duree de l'ensoleillement ainsi que
l'azimut du lever et du coucher.

Entrez la date d'observation sous la
forme JJ,MM,AAAA ? 1 ,5 ,1984
```

```
Entrez la longitude (comptee vers
l'est positivement et vers l'ouest
negativement) ? 5

Entrez la latitude ? 44

Pour la date consideree, en T.U.,
le soleil se levera a :
4 H 35 MN
et se couchera a :
18 H 38 MN
pour un ensoleillement total de
14 H 3 MN

Azimut du lever   : 68.4508684 DEG
Azimut du coucher: 291.549132 DEG

Voulez-vous faire un autre calcul ?
```

PRECESSION DES EQUINOXES

Ce phénomène connu depuis l'Antiquité est à l'origine du mouvement apparent des étoiles autour du pôle nord céleste ; ainsi, dans quelques milliers d'années, notre Etoile polaire sera Véga de la Lyre. Les coordonnées équatoriales d'un astre, c'est-à-dire son ascension droite et sa déclinaison, ne sont valables qu'à un instant donné. Les cartes astronomiques fournissent des coordonnées relatives au 1er janvier 195Ø à Ø heure. L'astronome doit donc évaluer la dérive des astres ; le programme proposé effectue ce calcul.

Utilisation

Les données à fournir sont le jour, le mois et l'année d'observation, l'ascension droite (en heures, minutes, secondes) et la déclinaison (en degrés, minutes, secondes) de l'astre.

Structure du programme

Lignes 1Ø-6Ø : initialisations - affichage.
Lignes 7Ø-1ØØ : introduction de la date et de l'ascension droite.
Lignes 11Ø-12Ø : début du calcul.
Lignes 13Ø-14Ø : introduction de la déclinaison.
Lignes 15Ø-2ØØ : suite du calcul.
Lignes 21Ø-26Ø : affichage des résultats.
Lignes 27Ø-31Ø : demande pour un autre calcul.
Sous-programme 2ØØØ-2Ø5Ø : conversion des heures et fraction d'heure (ou degrés et fraction de degré) en heures, minutes, secondes (ou degrés, minutes, secondes).
Sous-programme 3ØØØ-3Ø1Ø : conversion inverse.

Programme

```
 10 REM  PRECESSION
 20 CLS:PRINT " Ce programme calcule la
derive en"
 30 PRINT "ascension droite ainsi que l'
ecart en"
 40 PRINT "declinaison du a la precessio
n des"
 50 PRINT "equinoxes depuis les coordonn
ees de"
 60 PRINT "reference de 1950"
 70 PRINT:PRINT:PRINT " Entrez la date d
'observation sous la"
 80 INPUT "forme JJ,MM,AAAA ";J,M,AN
 90 PRINT " Entrez l'ascension droite de
 l'astre"
 100 INPUT "sous la forme H,MN,S ";H,MN,
S
 110 GOSUB 3000
 120 AD=Y*360/24
 130 PRINT " Entrez la declinaison de l'
astre"
 140 INPUT "sous la forme DEG,',''  ";H,M
N,S
 150 GOSUB 3000
 160 DE=Y
 165 NA=J/360+(M-1)/12+AN-1950
 170 DA=NA*(3.07327+1.33617*SIN(AD*PI/18
0)*TAN(DE*PI/180))/240
 180 ED=NA*(20.0468*COS(AD*PI/180))/3600
 190 AD=(AD+DA)*24/360
 200 DE=DE+ED
 210 PRINT:PRINT:PRINT "L'ascension droi
te est de :":PRINT
 220 Y=AD
 225 F=1
 230 GOSUB 2000
 240 PRINT:PRINT "La declinaison est de
:":PRINT
```

```
250 Y=DE
255 F=0
260 GOSUB 2000
270 PRINT:PRINT:PRINT "Voulez-vous fair
e un autre calcul ?"
280 GET K$
300 IF K$<>"N" GOTO 70
310 END
2000 H=INT(Y)
2005 Y=(Y-H)*60
2010 MN=INT(Y)
2020 S=INT((Y-MN)*60)
2030 IF F=1 THEN PRINT H;"H";MN;"MN";S;
"S"
2040 IF F=0 THEN PRINT H;"DEG";MN;"'";S
;"''"
2050 RETURN
3000 Y=H+MN/60+S/3600
3010 RETURN
```

Exemple

```
 Ce programme calcule la derive en
ascension droite ainsi que l'ecart en
declinaison du a la precession des
equinoxes depuis les coordonnees de
reference de 1950

 Entrez la date d'observation sous la
forme JJ,MM,AAAA ? 15 , 4 ,1984
 Entrez l'ascension droite de l'astre
sous la forme H,MN,S ? 18, 36 ,0
 Entrez la declinaison de l'astre
sous la forme DEG,',''  ? 38 ,49 , 0

L'ascension droite est de :

18 H 37 MN 8 S
```

La declinaison est de :

38 DEG 50 ' 47 ''

Voulez-vous faire un autre calcul ?

CONVERSION DE COORDONNEES ASTRONOMIQUES

Le programme suivant effectue la conversion des coordonnées astronomiques d'un astre (ascension droite et déclinaison) en azimut et hauteur.

Utilisation

Les données à introduire sont la date d'observation, la longitude (positive vers l'Est, négative vers l'Ouest) et la latitude de l'observateur (en degrés et fraction de degré), les coordonnées astronomiques de l'astre et l'heure d'observation en temps universel (TU), soit pour la France l'heure légale moins deux heures en été, moins une heure en hiver.

Le programme calcule alors l'azimut, compté positivement depuis le Nord dans le sens des aiguilles d'une montre (négativement sinon) de Ø° à 36Ø°; ainsi 9Ø° correspond à l'Est, 225° au Sud-Ouest, etc. Le programme indique aussi la hauteur, qui se compte depuis l'horizon pris comme Ø°, jusqu'au zénith qui correspond à 9Ø°.

Structure du programme

Lignes 1Ø-4Ø : initialisations - affichage.
Lignes 5Ø-2ØØ : introduction des données - conversions des angles.
Lignes 21Ø-27Ø : calcul de l'azimut et de la hauteur.
Lignes 28Ø-31Ø : affichage des résultats.
Lignes 32Ø-36Ø : demande pour un autre calcul.
Sous-programme 2ØØØ-2Ø5Ø : calcul du nombre de jours depuis le début de l'année en cours.

Sous-programme 3ØØØ-3Ø1Ø : conversion des heures, mn, s (ou °, ′, ″) en heures et fraction d'heure (ou ° et fraction de °).

Programme

```
 10 REM  CONVERSION
 15 PAPER 0:INK 2
 20 CLS:PRINT " Ce programme transforme
les coordon-"
 30 PRINT "nees astronomiques (ascension
 droite"
 40 PRINT "et declinaison) en azimut et
hauteur"
 50 PRINT:PRINT:PRINT " Entrez la date d
'observation sous la"
 60 INPUT "forme JJ,MM,AAAA ";J,M,AN
 70 PRINT:INPUT "Entrez votre longitude
";LO
 80 INPUT "Entrez votre latitude ";LA
 90 PRINT:PRINT "Entrez l'heure d'observ
ation (en T.U.)"
 100 INPUT "sous la forme H,MN,S ";H,MN,
S
 105 CO=PI/180
 110 GOSUB 3000
 120 HE=Y
 130 PRINT:PRINT "Entrez l'ascension dro
ite de l'astre"
 140 INPUT "sous la forme H,MN,S";H,MN,S

 150 GOSUB 3000
 160 AD=Y
 170 PRINT "Entrez la declinaison de l'a
stre"
 180 INPUT "sous la forme DEG,',''";H,MN
,S
 190 GOSUB 3000
 200 DE=Y
 210 GOSUB 2000
```

```
 220 T=6.617+0.065709*NJ+1.0022733*HE+LO
/15
 230 IF T>24 THEN T=T-24
 240 A=(T-AD)*15
 250 X=SIN(LA*CO)*SIN(DE*CO)+COS(LA*CO)*
COS(DE*CO)*COS(A*CO)
 260 HA=(ATN(X/SQR(1-X^2)))/CO
 270 AZ=(ATN(SIN(A*CO)/(COS(A*CO)*SIN(LA
*CO)-TAN(LA*CO)*COS(DE*CO))))/CO
 280 PRINT:PRINT:PRINT "L'astre se situe
 a l'azimut :"
 290 PRINT:PRINT AZ;"DEG":PRINT
 300 PRINT "et a la hauteur :"
 310 PRINT:PRINT HA;"DEG":PRINT
 320 PRINT:PRINT:PRINT "Voulez-vous fair
e un autre calcul ?"
 330 GET K$
 350 IF K$<>"N" GOTO 50
 360 END
 2000 IF M<=2 THEN AN=AN-1:M=M+13:GOTO 2
020
 2010 M=M+1
 2020 D=J+INT(365.25*AN)+INT(30.6001*M)
 2030 D0=INT(365.25*(AN-1))+428
 2040 NJ=D-D0
 2050 RETURN
 3000 Y=H+MN/60+S/3600
 3010 RETURN
```

Exemple

```
 Ce programme transforme les coordon-
nees astronomiques (ascension droite
et declinaison) en azimut et hauteur

 Entrez la date d'observation sous la
forme JJ,MM,AAAA ? 15, 4 , 1984
Entrez votre longitude ? 5
Entrez votre latitude ? 44
```

```
Entrez l'heure d'observation (en T.U.)
sous la forme H,MN,S ? 21 ,0, 0

Entrez l'ascension droite de l'astre
sous la forme H,MN,S ? 10 ,40 ,0
Entrez la declinaison de l'astre
sous la forme DEG,',''  ? 10 ,0 ,0

L'astre se situe a l'azimut :

-16.704458 DEG

et a la hauteur :

55.781959 DEG


Voulez-vous faire un autre calcul ?
```

TRI

Ce programme permet de classer dans l'ordre croissant une liste d'éléments numériques ou alphanumériques.

Utilisation

On doit d'abord indiquer si l'on veut trier des nombres ou des chaînes de caractères, puis le nombre d'éléments à trier.

On introduit ensuite la liste des éléments à trier, et on choisit d'afficher le résultat sur l'écran ou sur l'imprimante.

Le programme affiche alors la liste triée.

Structure du programme

Lignes 1Ø-3Ø : initialisations.
Lignes 4Ø-8Ø : choix du type d'éléments à trier.
Lignes 1ØØ-11Ø : introduction du nombre d'éléments N - dimensionnement du tableau T$ qui contiendra la liste triée.
Lignes 12Ø-24Ø : boucle principale du programme, qui se décompose ainsi :
Lignes 13Ø-14Ø : introduction du $i^{ème}$ élément, que l'on range dans une variable intermédiaire IN$.
Ligne 15Ø : remet à Ø la variable J qui contiendra le rang dans la liste triée de l'élément que l'on vient d'introduire.
Lignes 16Ø-19Ø : cette boucle secondaire s'arrête dès que l'élément que l'on vient d'introduire est inférieur (au sens numérique, ou bien au sens de l'ordre alphabétique, selon le cas) à un élément de la liste triée, ou bien quand on a examiné tous les éléments de la liste triée. Ceci permet de mettre J à la bonne valeur.
Lignes 2ØØ-22Ø : cette boucle secondaire permet de décaler les éléments de numéros J à I−1, qui deviennent ainsi les éléments de numéros J+1 à I.

Ligne 23Ø : l'élément que l'on vient d'introduire est placé au rang J de la liste triée.
Lignes 25Ø-33Ø : choix de l'affichage et affichage proprement dit.

Programme

```
 10 REM  TRI
 20 PAPER 0:INK 2
 30 CLS
 40 PRINT " Voulez-vous trier :"
 50 PRINT:PRINT "  1  Des nombres"
 60 PRINT:PRINT "  2  Des chaines alphan
umeriques"
 70 GET K$
 80 IF K$<"1" OR K$>"2" GOTO 70
 100 PRINT:INPUT " Nombre d'elements a t
rier ";N:PRINT
 110 DIM T$(N)
 120 FOR I=1 TO N
 130 PRINT "Element";I;
 140 INPUT IN$
 150 J=0
 160 REPEAT
 170 J=J+1
 180 F=(K$="1") AND(VAL(IN$)<VAL(T$(J)))
 OR (K$="2") AND (IN$<T$(J))
 190 UNTIL J=I OR F
 200 FOR K=I-1 TO J STEP -1
 210 T$(K+1)=T$(K)
 220 NEXT K
 230 T$(J)=IN$
 240 NEXT I
 250 CLS
 260 PRINT " Affichage sur :"
 270 PRINT:PRINT "  1  Ecran"
 280 PRINT:PRINT "  2  Imprimante"
 290 GET K$
 300 IF K$<"1" OR K$>"2" GOTO 290
 310 FOR L=1 TO N
```

```
 320 IF K$="1" THEN PRINT T$(L) ELSE LPR
INT T$(L)
 330 NEXT L
```

Exemple

```
 Voulez-vous trier :

  1  Des nombres
  2  Des chaines alphanumeriques

 Nombre d'elements a trier ? 4

Element 1 ? 2
Element 2 ? 1.145E-4
Element 3 ? -3.234
Element 4 ? 1.5434E+5

 Affichage sur :

  1  Ecran
  2  Imprimante

 -3.234
 1.145E-4
 2
 1.5434E+5

 Voulez-vous trier :

  1  Des nombres
  2  Des chaines alphanumeriques

 Nombre d'elements a trier ? 5

Element 1 ? ORIC
Element 2 ? ORDINATEUR
Element 3 ? IMPRIMANTE
```

```
Element 4 ? PROGRAMME
Element 5 ? MCP 40

 Affichage sur :

  1  Ecran
  2  Imprimante

 IMPRIMANTE
 MCP 40
 ORDINATEUR
 ORIC
 PROGRAMME
```

CARTE DE FRANCE

Ce programme affiche en haute résolution une carte de France.

Utilisation

Il suffit de lancer le programme ; la carte s'affiche immédiatement.

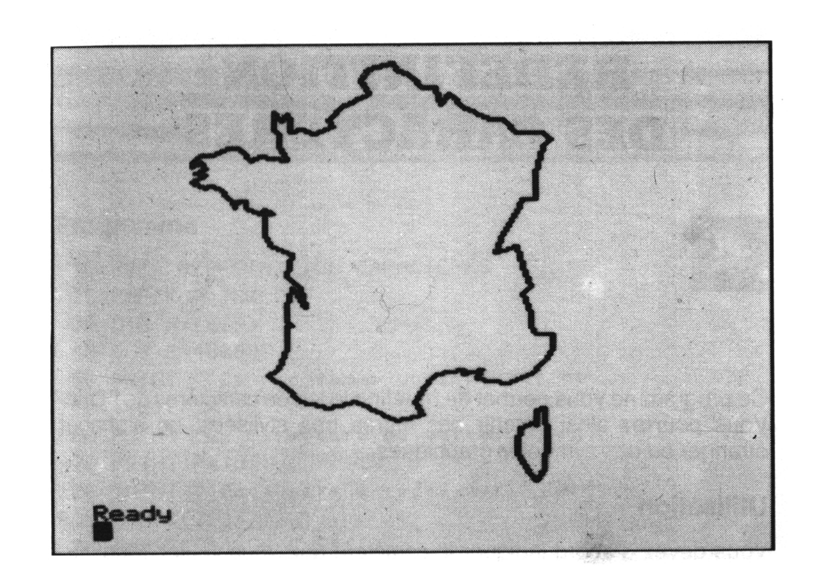
Structure du programme

Lignes 1Ø-2Ø : initialisations.
Ligne 3Ø : lecture des premières données.
Ligne 4Ø : positionnement du curseur.
Ligne 5Ø : lecture des données suivantes.
Lignes 6Ø-7Ø : tests ; si XB=Ø, la carte est finie ; si XB=999, il faut « lever la plume » pour aller dessiner la Corse.
Lignes 8Ø-1ØØ : permet le tracé au fur et à mesure de la carte.
Lignes 11Ø-44Ø : données utiles au programme.

Programme

```
10 REM  CARTE DE FRANCE
20 HIRES
30 READ XA,YA:YA=INT(YA*1.2)
40 CURSET XA-20,YA,1
50 READ XB,YB:YB=INT(YB*1.2)
60 IF XB=0 THEN END
70 IF XB=999 GOTO 30
80 DRAW XB-XA,YB-YA,1
90 XA=XB:YA=YB
100 GOTO 50
110 DATA 144,7,135,10,132,15,135,20
120 DATA 125,25,120,30,122,32,115,35
```

```
130 DATA 110,33,106,34,106,30,105,27
140 DATA 103,30,100,26,100,33,102,35
150 DATA 102,40,105,45,95,43,92,45
160 DATA 87,40,83,40,77,42,73,42,70,45
170 DATA 70,46,73,45,75,47,71,49,75,50
180 DATA 70,52,75,55,77,54,80,55,85,60
190 DATA 86,62,87,60,90,60,90,62,95,65
200 DATA 97,63,100,65,97,65,97,73
210 DATA 100,77,107,80,105,88,110,92
220 DATA 112,100,107,92,105,100
230 DATA 105,105,107,106,105,106
240 DATA 105,115,102,122,98,125
250 DATA 105,130,110,135,120,137
260 DATA 130,135,140,140,145,142
270 DATA 155,140,152,135,155,130
280 DATA 160,125,170,123,170,125
290 DATA 177,123,175,126,180,127
300 DATA 187,128,195,125,200,120
310 DATA 203,115,203,110,197,110
320 DATA 195,107,195,102,192,95,195,92
330 DATA 192,85,190,77,182,80,185,72
340 DATA 190,65,192,60,197,60,197,50
350 DATA 200,40,203,37,200,35,190,35
360 DATA 185,30,180,30,170,25,165,20
370 DATA 162,24,160,17,155,17,150,12
380 DATA 145,10,144,7
390 DATA 999,999
400 DATA 200,137,197,138,197,144
410 DATA 196,143,191,145,190,150
420 DATA 190,155,193,160,197,165
430 DATA 200,160,201,150,200,137
440 DATA 0,0
```


Ready

REDEFINITION DES CARACTERES



Ce programme vous permet de redéfinir le jeu de caractères de l'Oric. Vous pourrez ainsi obtenir des lettres très stylisées, un alphabet étranger ou des symboles graphiques.

Utilisation

Vous devez d'abord indiquer le caractère que vous voulez redéfinir. Puis une grille s'affiche, formée de points et d'étoiles aux endroits correspondant à un point lumineux du caractère. Un curseur clignotant indique votre position dans la grille. Vous pouvez le déplacer grâce aux quatre touches de déplacement du curseur. Pour effacer un point, appuyez sur « Ø » (un petit point s'affiche alors à l'endroit indiqué par le curseur) ; pour allumer un point, appuyez sur « 1 » (cette fois-ci, une étoile s'affiche). Appuyez sur la touche « Return » lorsque vous avez fini.

Structure du programme

Lignes 1Ø-7Ø : initialisations - affichage.
Lignes 8Ø-9Ø : introduction du caractère à redéfinir.
Lignes 1ØØ-14Ø : remplit le tableau M% avec les valeurs correspondant au dessin du caractère.
Lignes 15Ø-21Ø : affichage de la grille.
Lignes 22Ø-23Ø : affiche le curseur et définit ses coordonnées initiales.
Ligne 24Ø : affiche le caractère à redéfinir.
Lignes 31Ø-41Ø : attend que l'utilisateur appuie sur une touche, puis effectue le traitement approprié : modification du tableau M% ou bien déplacement du curseur, et ainsi de suite jusqu'à ce

que l'utilisateur appuie sur « Return ».

Lignes 42Ø-43Ø : rétablit le curseur normal et retourne en 8Ø pour redéfinir un autre caractère.

Sous-programme 1ØØØ-1Ø8Ø : place dans la mémoire de l'Oric les valeurs nécessaires pour que le caractère soit modifié.

Programme

```
 10 REM  REDEFINITION CARACTERES
 20 PAPER 0:INK 2
 30 DIM M%(5,7)
 40 CLS:B=46080
 50 PRINT " Ce programme vous permet de
redefinir";
 60 PRINT "un caractere de votre choix."
 70 PRINT:PRINT
 80 INPUT " Caractere a redefinir ";C$:P
RINT CHR$(17)
 90 C=ASC(C$):C$=CHR$(C)
 100 FOR I=0 TO 7
 110 FOR J=0 TO 5
 120 M%(J,I)=-((PEEK(C*8+B+I) AND 2^(5-J
))<>0)
 130 NEXT J
 140 NEXT I
 150 CLS
 160 FOR I=0 TO 7
 170 FOR J=0 TO 5
 180 IF M%(J,I)=0 THEN PRINT @ 5+J*2,5+I
*2;"." ELSE PRINT @ 5+J*2,5+I*2;"*"
 190 NEXT J
 200 PRINT @ 2,6+I*2;CHR$(27);"L"
 210 NEXT I
 220 PRINT @ 5,6;"^"
 230 X=0:Y=0
 240 PRINT @ 30,9;C$
 310 REPEAT
 320 GET K$
 330 IF K$="0" THEN F=(M%(X,Y)<>0):IF F
THEN M%(X,Y)=0:PRINT @ 2*X+5,2*Y+5;"."
```

```
 340 IF K$="1" THEN F=(M%(X,Y)<>1):IF F
THEN M%(X,Y)=1:PRINT @ 2*X+5,2*Y+5;"*"
 350 IF K$=CHR$(11) THEN PRINT @ 2*X+5,2
*Y+6;" ":Y=Y-1:IF Y<0 THEN Y=7
 360 IF K$=CHR$(8) THEN PRINT @ 2*X+5,2*
Y+6;" ":X=X-1:IF X<0 THEN X=5
 370 IF K$=CHR$(9) THEN PRINT @ 2*X+5,2*
Y+6;" ":X=X+1:IF X>5 THEN X=0
 380 IF K$=CHR$(10) THEN PRINT @ 2*X+5,2
*Y+6;" ":Y=Y+1:IF Y>7 THEN Y=0
 390 PRINT @ 2*X+5,2*Y+6;"^"
 400 IF F THEN GOSUB 1000
 410 UNTIL K$=CHR$(13)
 420 PRINT @ 2,22;CHR$(17)
 430 GOTO 80
 1000 FOR I=0 TO 7
 1010 P=0
 1020 FOR J=0 TO 5
 1030 P=P+M%(J,I)*2^(5-J)
 1040 NEXT J
 1050 POKE B+C*8+I,P
 1060 NEXT I
 1070 F=0
 1080 RETURN
```

```
                                        CAPS

* * * * * .
* . . . . .
* . . . . .              C
* . . . . .
* . . . . .
* . . . . .
* * * * * .
↑
. . . . . .

Caractere a redefinir ?
```

DESASSEMBLEUR

Ce programme permet d'afficher et de désassembler le contenu de n'importe quelle partie de la mémoire de l'Oric.

Utilisation

Il faut préciser les adresses de départ et de fin en hexadécimal.

On doit ensuite indiquer si l'on veut que le programme affiche des mnémoniques (désassemblage proprement dit) ou des codes ASCII, et si l'on veut afficher sur l'écran ou sur l'imprimante.

Remarque : les codes correspondant aux instructions-machine proprement dites sont affichés sous forme standard (ex. : LDA, JSR, RTS...) ; par contre, les modes d'adressage sont affichés sous une forme directe qui nous semble plus claire :

IMM : adressage immédiat.
ABS : adressage absolu.
ABX : adressage absolu, indexé par X.
ABY : adressage absolu, indexé par Y.
ZPG : adressage en page zéro.
ZPX : adressage en page zéro, indexé par X.
ZPY : adressage en page zéro, indexé par Y.
IND : adressage indirect.
IDX : adressage indirect, indexé par X.
IDY : adressage indirect, indexé par Y.

(Les autres modes d'adressage, à savoir l'adressage implicite et l'adressage relatif, sont représentés par une chaîne de caractères vide.)

Structure du programme

Lignes 1Ø-1ØØ : initialisations.
Lignes 11Ø-15Ø : introduction des données - conversion en décimal.

Ligne 16Ø : teste si l'on doit afficher des codes ASCII.

Lignes 19Ø-27Ø : c'est le désassemblage proprement dit, on va chercher le contenu d'une adresse, on regarde à quel code-machine il correspond et on appelle le sous-programme réalisant le traitement approprié.

Lignes 3ØØ-36Ø : cette partie affiche les codes ASCII des cases-mémoire demandées.

Sous-programmes 1ØØØ, 2ØØØ, 3ØØØ, 4ØØØ : selon le mode d'adressage de l'instruction, un traitement différent est effectué.

Sous-programme 5ØØØ-5Ø3Ø : conversion de décimal en hexadécimal.

Sous-programme 55ØØ-553Ø : conversion d'hexadécimal en décimal.

Sous-programme 6ØØØ-6Ø4Ø : initialisations.

Sous-programme 61ØØ-611Ø : imprime sur l'écran ou l'imprimante selon le cas.

Lignes 62ØØ-64ØØ : données utiles au programme.

Programme

```
 10 REM  DESASSEMBLEUR
 20 PAPER 0:INK 2
 30 CLS
 100 GOSUB 6000
 110 INPUT "Adresse de depart ($) ";D$
 120 INPUT "Adresse de fin    ($) ";F$
 130 H$=D$:GOSUB 5500:D=H:H$=F$:GOSUB 55
00:F=H
 133 PRINT "Mnemoniques (1) ou ASCII (2)
 ?"
 137 GET H$:P=VAL(H$):IF P<>1 AND P<>2 T
HEN GOTO 137
 140 PRINT "Ecran (1) ou imprimante (2)
?"
 150 GET H$:Q=VAL(H$):IF Q<>1 AND Q<>2 T
HEN GOTO 150
 160 IF P=2 THEN GOTO 300
 190 I=D
 200 P=PEEK(I):MN$=MN$(VAL(LEFT$(C$(P),2
)))
 210 AD=ASC(RIGHT$(C$(P),1))-65:AD$=AD$(
AD)
```

```
 220 H=I:N=4:GOSUB 5000:Q$=".    "+H$:X=0
:GOSUB 6100
 230 H=P:N=2:GOSUB 5000:Q$=" "+H$:X=0:GO
SUB 6100
 250 ON AD+1 GOSUB 1000,2000,2000,2000,2
000,2000,2000,3000,3000,3000,3000,4000
 260 IF I<F THEN I=I+1:GOTO 200
 270 GOTO 110
 300 I=D
 310 P=PEEK(I):IF P<32 OR P>125 AND P<16
0 THEN P$=""ELSEP$=CHR$(P+128*(P>159))
 320 H=I:N=4:GOSUB 5000:Q$=".    "+H$:X=0
:GOSUB 6100
 330 H=P:N=2:GOSUB 5000:Q$=" "+H$:X=0:GO
SUB 6100
 340 Q$=" "+P$:X=1:GOSUB 6100
 350 IF I<F THEN I=I+1:GOTO 310
 360 GOTO 110
 1000 Q$="           "+MN$:X=1:GOSUB 6100
 1010 RETURN
 2000 I=I+1:H=PEEK(I):GOSUB 5000:Q$=" "+
H$+"     "+MN$+" "+AD$:X=0:GOSUB 6100
 2010 IF AD<>1 THEN H$="00"+H$
 2020 Q$=" "+H$:X=1:GOSUB 6100
 2030 RETURN
 3000 I=I+1:H=PEEK(I):GOSUB 5000:Q$=" "+
H$:X=0:GOSUB 6100:HH$=H$
 3010 I=I+1:H=PEEK(I):GOSUB 5000:Q$=" "+
H$:X=0:GOSUB 6100:HH$=H$+HH$
 3020 Q$="  "+MN$+" "+AD$+" "+HH$:X=1:GO
SUB 6100
 3030 RETURN
 4000 I=I+1:P=PEEK(I):H=P:GOSUB 5000:Q$=
" "+H$+"     "+MN$:X=0:GOSUB 6100
 4010 P=I+1+P+256*(P>127):N=4:H=P:GOSUB
5000:Q$="      "+H$:X=1:GOSUB 6100
 4020 RETURN
 5000 H$="":FOR K=1 TO N:T=INT(H/16)
 5010 H=H-16*T:H=48+H-7*(H>9)
 5020 H$=CHR$(H)+H$:H=T:NEXT
```

```
 5030 RETURN
 5500 H=0:FOR K=1 TO 4
 5510 T=ASC(MID$(H$,K,1))-48:T=T+7*(T>9)
 5520 H=H+T*16^(4-K):NEXT
 5530 RETURN
 6000 DIM AD$(11),MN$(56),C$(255)
 6010 FOR I=0 TO 11:READ AD$(I):NEXT
 6020 FOR I=0 TO 56:READ MN$(I):NEXT
 6030 FOR I=0 TO 255:READ C$(I):NEXT
 6040 RETURN
 6100 IF Q=1 THEN PRINT Q$;:IF X=1 THEN
PRINT
 6105 IF Q=2 THEN LPRINT Q$;:IF X=1 THEN
 LPRINT
 6110 RETURN
 6200 DATA "",IMM,ZPG,ZPX,ZPY,IDX,IDY,AB
S,ABX,ABY,IND,"",???
 6210 DATA ADC,AND,ASL,BCC,BCS,BEQ,BIT,B
MI,BNE,BPL,BRK,BVC,BVS,CLC
 6220 DATA CLD,CLI,CLV,CMP,CPX,CPY,DEC,D
EX,DEY,EOR,INC,INX,INY,JMP
 6230 DATA JSR,LDA,LDX,LDY,LSR,NOP,ORA,P
HA,PHP,PLA,PLP,ROL,ROR,RTI
 6240 DATA RTS,SBC,SEC,SED,SEI,STA,STX,S
TY,TAX,TAY,TSX,TXA,TXS,TYA
 6250 DATA 11A,35F,00A,00A,00A,35C,03C,0
0A,37A,35B,03A,00A,00A,35H,03H,00A
 6260 DATA 10L,35G,00A,00A,00A,35D,03D,0
0A,14A,35J,00A,00A,00A,35I,03I,00A
 6270 DATA 29H,02F,00A,00A,07C,02C,40C,0
0A,39A,02B,40A,00A,07H,02H,40H,00A
 6280 DATA 08L,02G,00A,00A,00A,02D,40D,0
0A,45A,02J,16A,00A,00A,02I,40I,00A
 6290 DATA 42A,24F,00A,00A,00A,24C,33C,0
0A,36A,24B,33A,00A,28H,24H,33H,00A
 6300 DATA 12L,24G,00A,00A,00A,24D,33D,0
0A,16A,24J,00A,00A,00A,24I,33I,00A
 6310 DATA 43A,01F,00A,00A,00A,01C,41C,0
0A,38A,01B,41A,00A,28K,01H,41H,00A
```

```
 6320 DATA 13L,01G,00A,00A,00A,01D,41D,0
0A,47A,01J,00A,00A,00A,01I,41I,00A
 6330 DATA 00A,48F,00A,00A,50C,48C,49C,0
0A,23A,00A,54A,00A,50H,48H,49H,00A
 6340 DATA 04L,48G,00A,00A,50D,48D,49E,0
0A,56A,48J,55A,00A,00A,48I,00A,00A
 6350 DATA 32B,30F,31B,00A,32C,30C,31C,0
0A,52A,30B,51A,00A,32H,30H,31H,00A
 6360 DATA 05L,30G,00A,00A,32D,30D,31E,0
0A,17A,30J,53A,00A,32I,30I,31J,00A
 6370 DATA 20B,18F,00A,00A,20C,18C,21C,0
0A,27A,18B,22A,00A,20H,18H,21H,00A
 6380 DATA 09L,18G,00A,00A,00A,18D,21D,0
0A,15A,18J,00A,00A,00A,18I,21I,00A
 6390 DATA 19B,44F,00A,00A,19C,44C,25C,0
0A,26A,44B,34A,00A,19H,44H,25H,00A
 6400 DATA 06L,44G,00A,00A,00A,44D,25D,0
0A,46A,44J,00A,00A,00A,44I,25I,00A
```

Exemple

```
Adresse de depart ($) ? FAE1
Adresse de fin    ($) ? FB05
Mnemoniques (1) ou ASCII (2) ?
Ecran (1) ou imprimante (2) ?
   FAE1 A2 06      LDX IMM 06
   FAE3 A0 FB      LDY IMM FB
   FAE5 20 86 FA   JSR ABS FA86
   FAE8 A9 00      LDA IMM 00
   FAEA AA         TAX
   FAEB 8A         TXA
   FAEC 48         PHA
   FAED A9 00      LDA IMM 00
   FAEF 20 90 F5   JSR ABS F590
   FAF2 A2 00      LDX IMM 00
   FAF4 CA         DEX
   FAF5 D0 FD      BNE     FAF4
   FAF7 68         PLA
   FAF8 AA         TAX
```

```
.   FAF9 E8          INX
.   FAFA E0 70       CPX IMM 70
.   FAFC D0 ED       BNE     FAEB
    FAFE A9 08       LDA IMM 08
.   FB00 A2 00       LDX IMM 00
.   FB02 20 90 F5    JSR ABS F590
.   FB05 60          RTS


Adresse de depart ($) ? C11F
Adresse de fin    ($) ? C135
Mnemoniques (1) ou ASCII (2) ?
Ecran (1) ou imprimante (2) ?
    C11F 55 U
.   C120 4E N
.   C121 54 T
.   C122 49 I
.   C123 CC L
.   C124 46 F
.   C125 4F O
.   C126 D2 R
.   C127 4C L
.   C128 4C L
.   C129 49 I
.   C12A 53 S
.   C12B D4 T
.   C12C 4C L
.   C12D 50 P
.   C12E 52 R
.   C12F 49 I
.   C130 4E N
.   C131 D4 T
.   C132 4E N
.   C133 45 E
.   C134 58 X
.   C135 D4 T
```

Annexe : Répertoire des commandes utilisées

	ABS	AND	ASC	ATN	CHR$	COS	CURSET	DEF, FN	DIM	DRAW	END	EXP	EXPLODE	FOR, TO, NEXT	STEP	GET	GOSUB, RETURN	GOTO	HIRES	IF, (THEN)	ELSE	INPUT	INT	KEY$	LEFT$	LEN	LIST	LN
Loto					●									●				●		●			●					
Cadavres exquis											●			●		●	●	●		●			●					
Clavier-réflexe					●											●		●		●			●	●				
Bataille									●		●			●	●	●		●		●			●		●	●		
Le nombre mystérieux														●		●		●		●	●	●	●					
La tour infernale	●	●											●			●		●		●		●	●					
Master Mind									●		●			●	●	●	●	●		●	●	●	●		●	●		
Les pois sauteurs	●	●							●					●		●		●		●		●						
Rayon X		●												●		●		●		●	●	●	●					
Bataille navale		●	●		●				●		●		●	●	●	●		●		●		●	●		●	●		
Simon		●			●						●			●		●	●	●		●			●			●		
Chenille		●			●				●		●		●	●			●	●		●	●		●					
Chiffres romains																●		●		●		●						
Nombre de jours entre deux dates	●										●					●	●	●		●		●	●					
Message					●															●		●		●		●		
Calendrier														●		●		●		●	●	●	●					
Calculatrice	●			●		●						●					●		●	●	●	●	●					●
Nombres premiers	●													●	●			●		●		●	●					

Annexe : Répertoire des commandes utilisées	ABS	AND	ASC	ATN	CHR$	COS	CURSET	DEF, FN	DIM	DRAW	END	EXP	EXPLODE	FOR, TO, NEXT	STEP	GET	GOSUB, RETURN	GOTO	HIRES	IF, (THEN)	ELSE	INPUT	INT	KEY$	LEFT$	LEN	LIST	LN
Décomposition en facteurs premiers	●																	●		●		●	●					
Factorielles	●													●				●		●		●	●			●		●
Changement de base			●		●									●				●		●		●	●			●		
Balistique						●												●		●		●						
Résolution par dichotomie						●		●								●		●		●		●					●	
Intégration	●							●				●		●	●	●		●		●		●	●				●	
Equation du second degré																●		●		●		●						
Courbes		●					●	●		●						●		●	●	●		●	●				●	
Calendriers Julien et Grégorien	●															●		●		●		●	●					
Phases de la lune											●					●	●	●		●	●	●	●					
Lever et coucher du soleil				●		●					●					●	●	●		●		●	●					
Précession des équinoxes																●	●			●		●	●					
Conversion de coordonnées astronomiques				●		●					●					●	●	●		●		●	●					
Tri		●							●					●	●	●		●		●	●	●						
Carte de France							●			●	●							●	●	●			●					
Redéfinition des caractères		●	●		●				●					●		●	●	●		●	●	●						
Désassembleur		●	●		●				●					●		●	●	●		●	●	●	●		●			

Annexe : Répertoire des commandes utilisées

	LOG	LPRINT	MID$	MUSIC	ON	OR	PEEK	PI	PING	PLOT PLOT	POKE	PRINT @	READ, DATA	REPEAT, UNTIL	RESTORE	RIGHT$	RND	SCRN	SGN	SIN	SQR	STR$	TAB	TAN	TEXT	VAL	WAIT	ZAP
Loto												●					●											
Cadavres exquis					●							●					●											
Clavier-réflexe												●		●			●										●	
Bataille												●	●				●						●				●	
Le nombre mystérieux						●							●		●		●									●		
La tour infernale						●								●			●						●				●	
Master Mind			●			●						●				●	●					●				●		
Les pois sauteurs						●				●				●														
Rayon X																	●											
Bataille navale						●						●				●	●										●	●
Simon			●	●	●							●		●		●	●					●				●	●	
Chenille							●		●	●		●					●	●				●						●
Chiffres romains													●	●	●													
Nombre de jours entre deux dates																												
Message			●									●		●		●											●	
Calendrier		●											●										●					
Calculatrice	●				●			●				●							●	●	●			●		●		
Nombres premiers																					●							

Annexe : Répertoire des commandes utilisées	LOG	LPRINT	MID$	MUSIC	ON	OR	PEEK	PI	PING	PLOT PLOT	POKE	PRINT @	READ, DATA	REPEAT, UNTIL	RESTORE	RIGHT$	RND	SCRN	SGN	SIN	SQR	STR$	TAB	TAN	TEXT	VAL	WAIT	ZAP
Décomposition en facteurs premiers																					●							
Factorielles								●								●						●						
Changement de base			●																									
Balistique								●												●	●							
Résolution par dichotomie						●								●														
Intégration						●																						
Equation du second degré																			●		●							
Courbes						●								●											●			
Calendriers Julien et Grégorien																												
Phases de la lune																												
Lever et coucher du soleil								●												●	●			●				
Précession des équinoxes								●												●				●				
Conversion de coordonnées astronomiques								●												●	●			●				
Tri		●				●								●												●		
Carte de France													●															
Redéfinition des caractères							●				●	●		●														
Désassembleur		●	●		●	●	●						●			●										●		

COLLECTION E.T.S.F. MICRO-SYSTEMES

1 - A. VILLARD et M. MIAUX, *Un microprocesseur pas à pas*
2 - A. VILLARD et M. MIAUX, *Systèmes à microprocesseur*
3 - P. GUEULLE, *Maîtrisez votre ZX 81*
4 - E. FLOEGEL, *Du Basic au Pascal*
5 - P. COURBIER, *Vous avez dit Basic ?*
6 - M. MARCHAND, *Vous avez dit micro ?*
7 - P. GUEULLE, *Pilotez votre ZX 81*
8 - M. JACQUELIN, *La micro-informatique et son A-B-C*
9 - M. OURY, *Maîtrisez les TO 7 et TO 7-70*
10 - P. GUEULLE, *Pilotez votre Oric 1 + Atmos*
11 - P. JOUVELOT et D. LE CONTE DES FLORIS, *Système d'exploitation et logiciel de base des micro-ordinateurs*
12 - P. GUEULLE, *Robotisez votre ZX 81*
13 - M. CAUT, *J'apprends le Basic*
14 - C. MALOSSE, C. TASSET, P. PRUT, *La micro, c'est pas sorcier !*
15 - R. GREGOIRE, *Bus IEEE*
16 - M. OURY, *Maîtrisez le MO5*
17 - P. GUEULLE, *Votre ordinateur et la télématique*
18 - P. COURBIER, *Connaissez-vous Macintosh ?*
19 - M. ROUSSELET, *Graphismes en kits*
20 - S. ARQUIE, *Micro-informatique et P.M.E.*
21 - R. SCHULZ, *60 solutions pour Oric 1 et Atmos*
22 - H. HUNIC, *Listes et tableaux en Basic*

Achevé d'imprimer
sur les presses
de l'Imprimerie Marcel Bon
70001 Vesoul
Dépôt légal : décembre 1984
N° éditeur : 445 - N° imprimeur : 2862

MICRO SYSTEMES

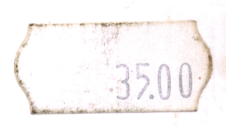


35 PROGRAMMES POUR ORIC 1 et ATMOS

Cet ouvrage familiarisera le lecteur, débutant ou averti, au **maniement des commandes Basic** des Oric 1 et Atmos.
35 programmes bien structurés abordent **des domaines variés :** jeux, vie pratique, mathématiques, astronomie, utilitaires. Ils peuvent être utilisés tels quels ou servir, plus ou moins modifiés, de point de départ ou de sous-programmes à des ensembles plus importants.

Quelques programmes :

- Cadavres exquis
- Chenille
- La tour infernale
- Pois sauteurs
- Calculatrice
- Factorielles
- Courbes
- Coordonnées astronomiques
- Redéfinition des caractères
- Désassemblage



ISBN 2-85535-100-6

TARIF CODE B

17

35 PROGRAMMES POUR ORIC 1 ET ATMOS D. LASSERAN